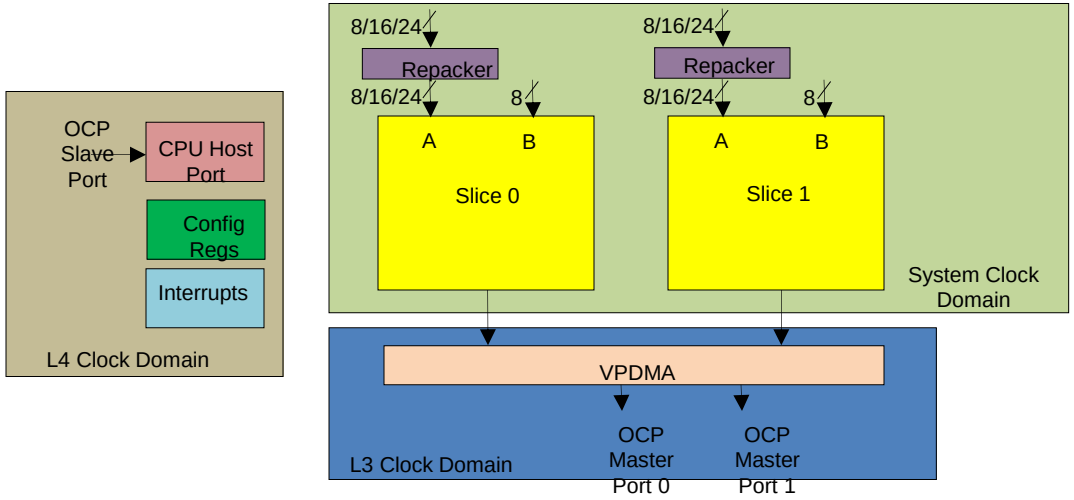
VIP Overview (Video Input Port)

9th March 2017
Version 1.0

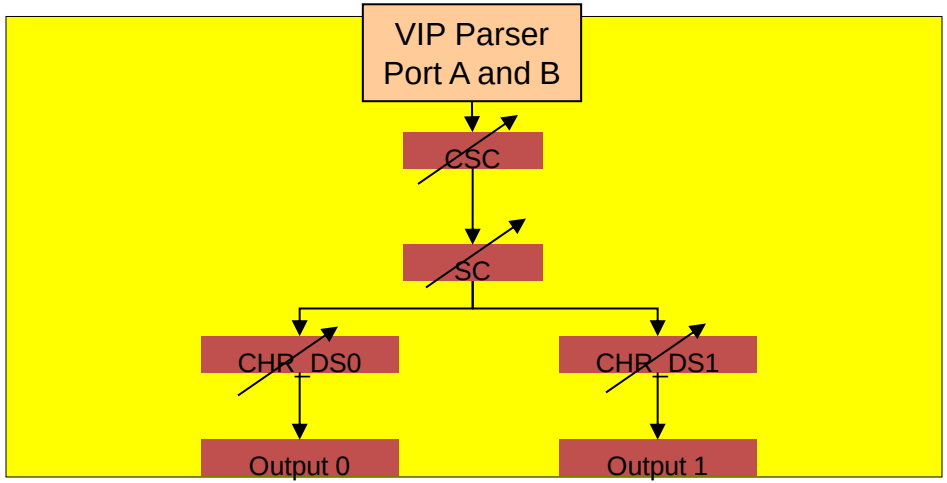
Agenda

- VIP Introduction
  - Features
  - SC
  - CSC
  - CHR_DS
  - VIP Capture data flows
- VIP Driver Overview
  - FVID2 Interface
  - Driver Design

VIP Sub-sytem Block



VIP Slice Block



VIP Block

- TDA2xx/AM572x/DRA72x Supports 3 VIP Block with two ports Port A and Port B per slice
  - VIP1/ VIP2 parser can operate as one 8/16/24 bit input port (Port A) or two 8-bit ports (Port A, Port B).
  - VIP3 parser can operate as 16 bit input port (Port A)
- Separate pixel clock and framing signals for each port
- Each VIP block has two slices : VIPx S0, VIPx S1
- VIP Block is used to capture video data from external video sources like video decoders or sensors

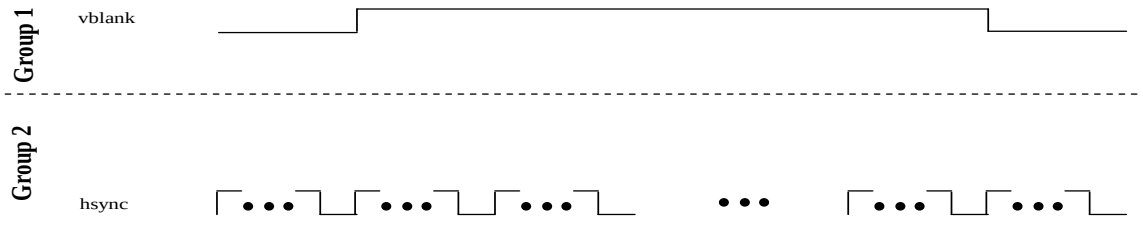
VIP Features

- Format conversion
  - Inputs: YUV422I, YUV444, RGB888
  - Outputs: YUV422I, YUV420SP (Uses CHR_DS), YUV422SP, RGB888 (Uses CSC)
- Supports optional scaling from 1/8x to 2048 pixels, only down scaling supported
- Supports optional color space conversion
  - YUV to RGB color space conversion using CSC block
- Supports optional chroma downsampler
  - YUV422I to YUV420SP conversion using CHR_DS
- Supports up to 165 MHz clock (includes blanking)
- Video Interface width
  - 8/16/24 bit mode
  - Combination: PortA 8-bit and PortB 8-bit or PortA 16 or 24-bit

Contd..

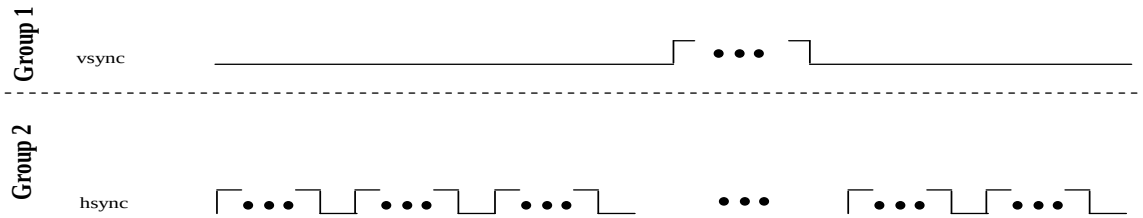
- Supports Embedded (BT.656/BT.1120 16/24b, BT.656 8b) or discrete (BT.601 style) sync
- Video Interface Mode
  - Discrete Sync
    - Single Channel non multiplexed mode with HSYNC and VBLK as control signals
    - Single Channel non multiplexed mode with HSYNC and VSYNC as control signals
    - Single Channel non multiplexed mode with AVID and VBLK as control signals
    - Single Channel non multiplexed mode with AVID and VSYNC as control signals
  - Embedded Sync
    - Single Channel non multiplexed mode
    - Multi-Channel pixel or line multiplexed mode

VIP Interface modes



- Active lines captured
- All pixels in line captured
- Frame Start: VBLK high to low
- Line start: HSYNC low to high

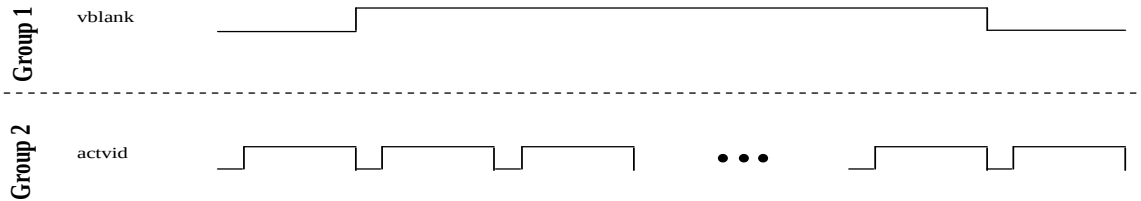
DS : Single Channel non multiplexed mode with HSYNC and VBLK as control signals



- All lines captured
- All pixels in line captured
- Frame Start: VSYNC low to high
- Line start: HSYNC low to high

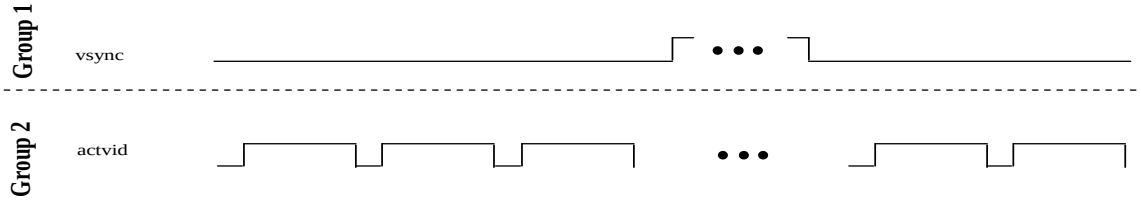
DS :Single Channel non multiplexed mode with HSYNC and VSYNC as control signals

Contd..



- Active lines captured
- Active pixels in line captured
- Frame Start: VBLK high to low
- Line start: ACTVID low to high

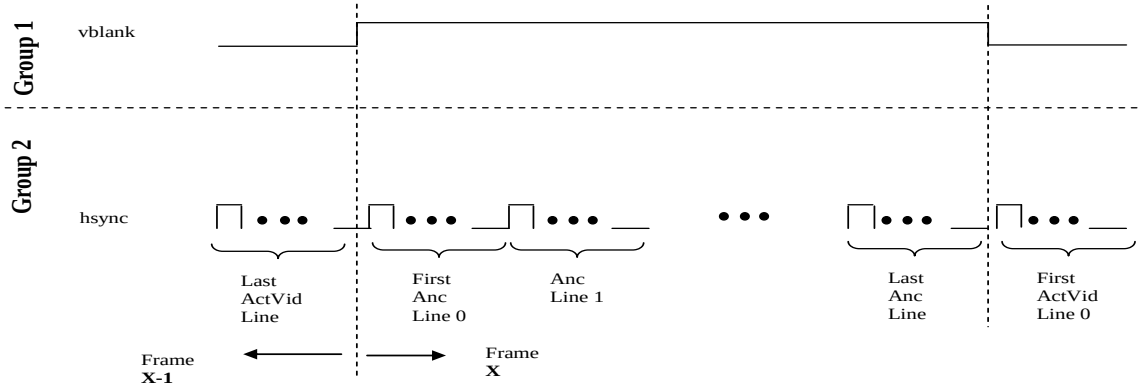
DS : Single Channel non multiplexed mode with AVID and VBLK as control signals



- All lines captured which has ACTVID high
- Active pixels in line captured
- Frame Start: VSYNC low to high
- Line start: ACTVID low to high

DS : Single Channel non multiplexed mode with AVID and VSYNC as control signals

Contd

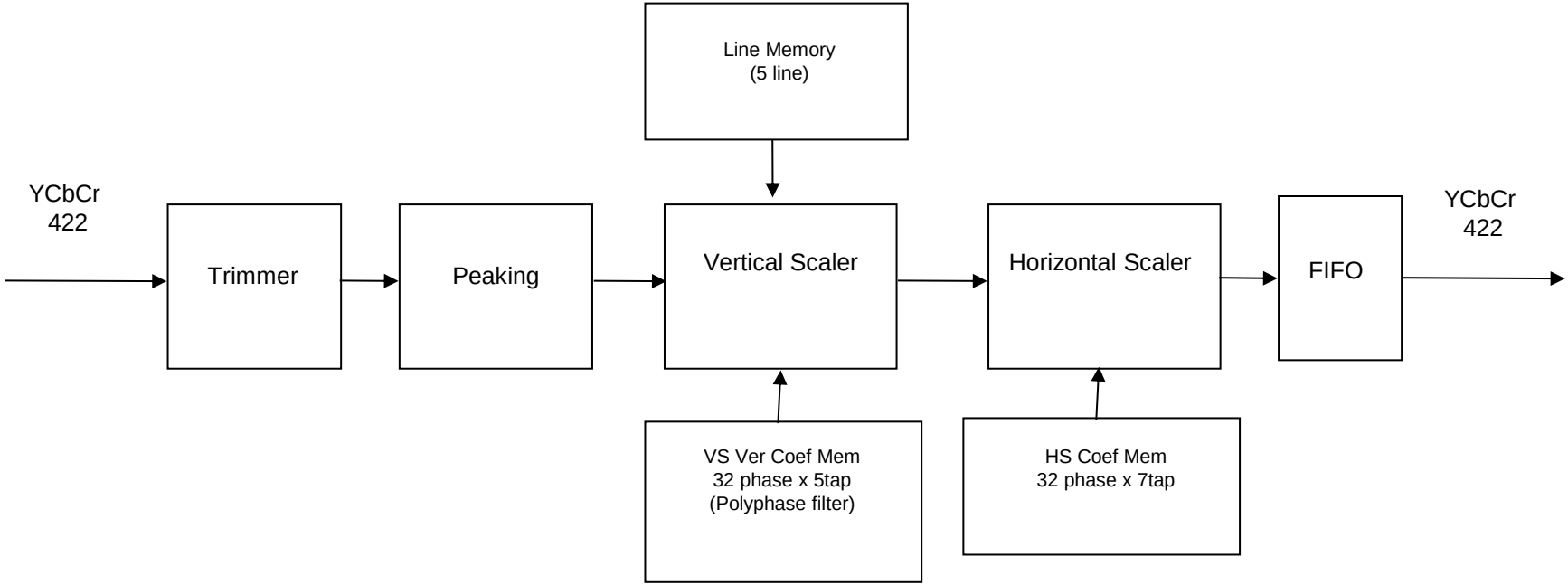


ES : Single Channel non multiplexed mode

SC Features

- Vertical and horizontal up and down scaling
- Polyphase filter upscaling
- Running average vertical down scaling
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support
- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- Scaling filter Coefficient memory download

SC Block



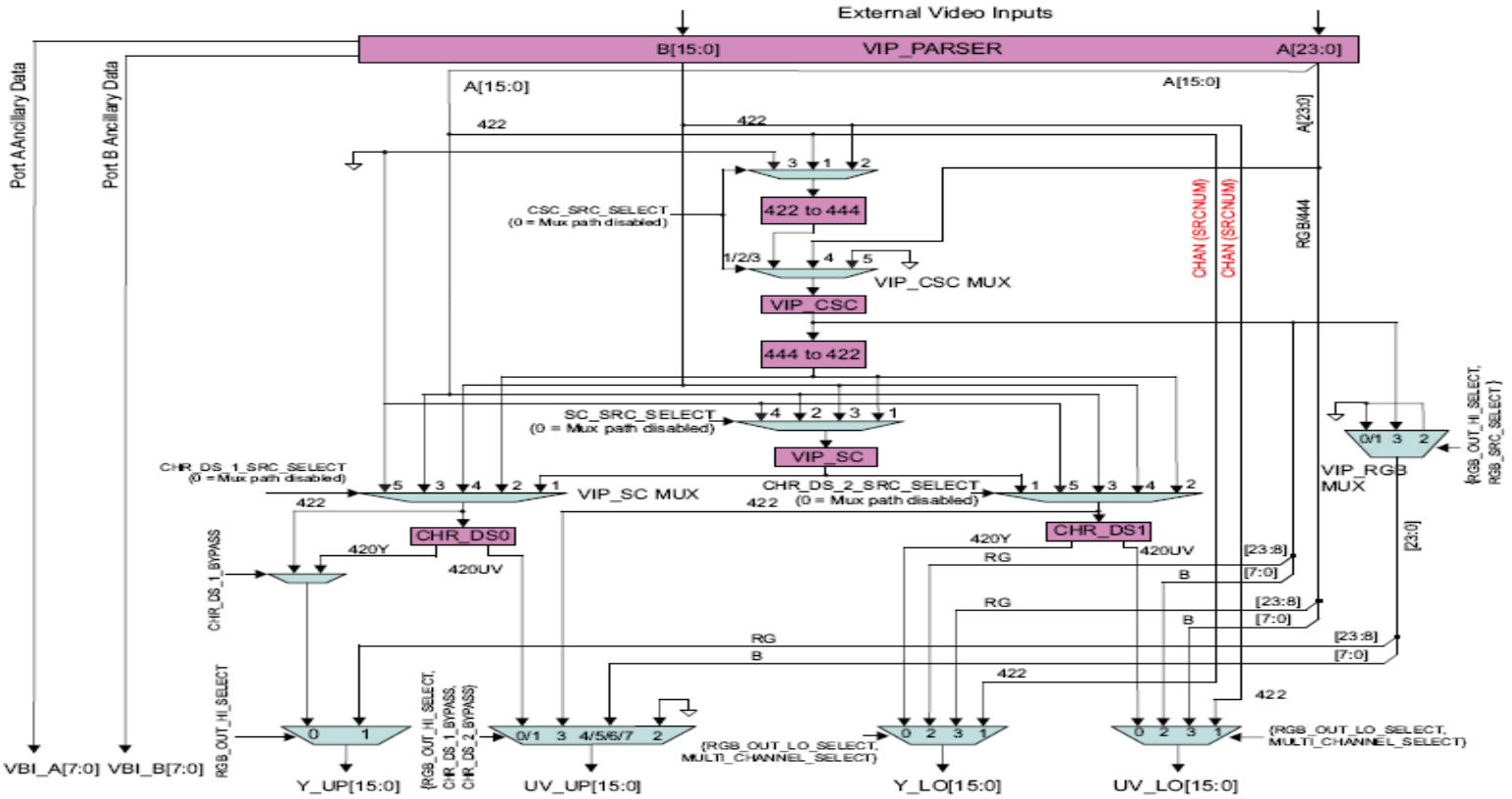
CSC Features

- A fully programmable color space converter. With the programmability, input video data in any color space can be converted to another color space.
- It could convert YCbCr to RGB and vice versa.
- This module could be put by pass as well if no conversion is required.

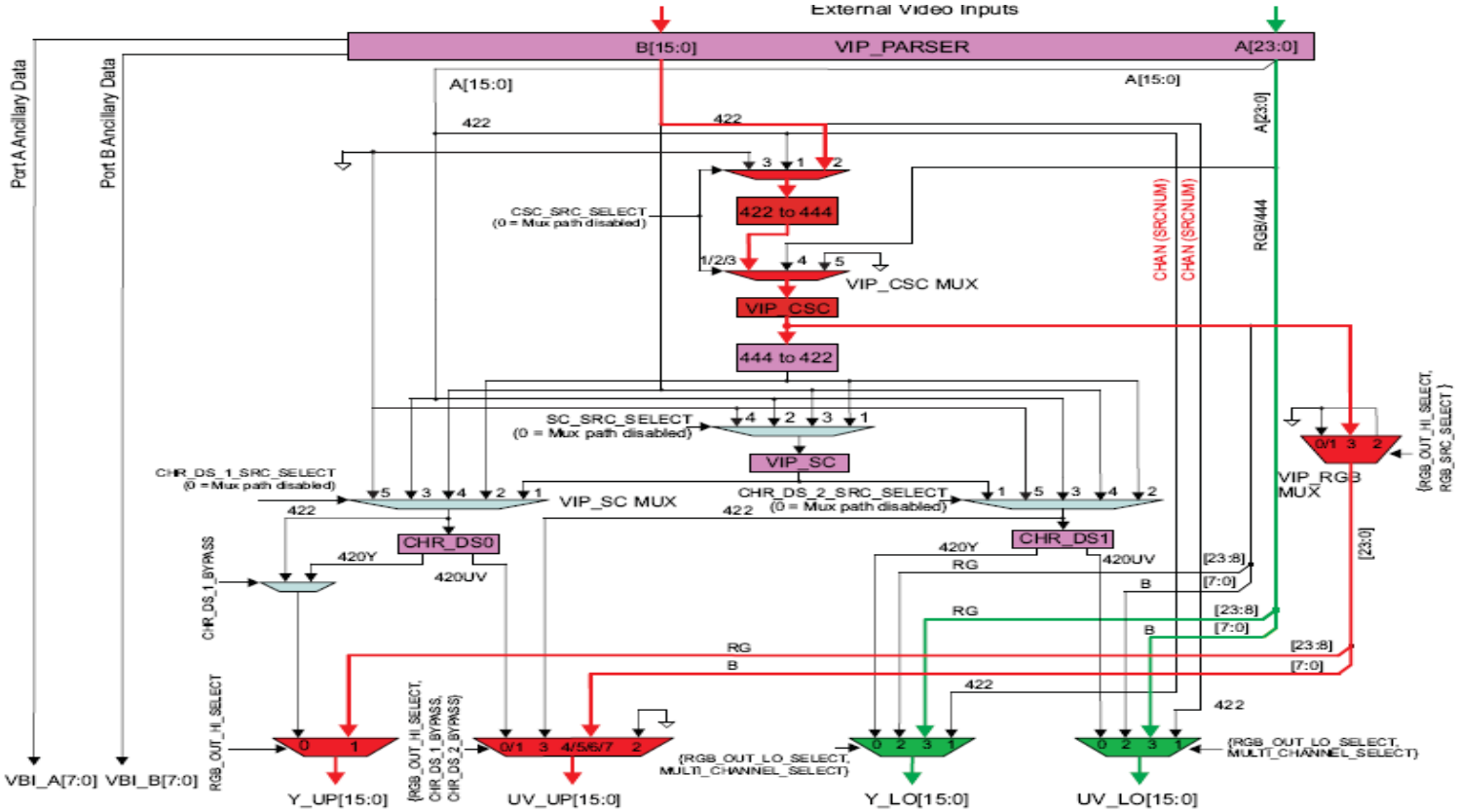
CHR_DS Features

- It is used to downsample picture input in the format 4:2:2 to 4:2:0
- This downsampling is required because typical video encoders expects input in 4:2:0 format before compression
- Down sampling is performed using averaging filter.

VIP Slice Detailed Block Diagram



Example Single Channel RGB / YUV422 Capture

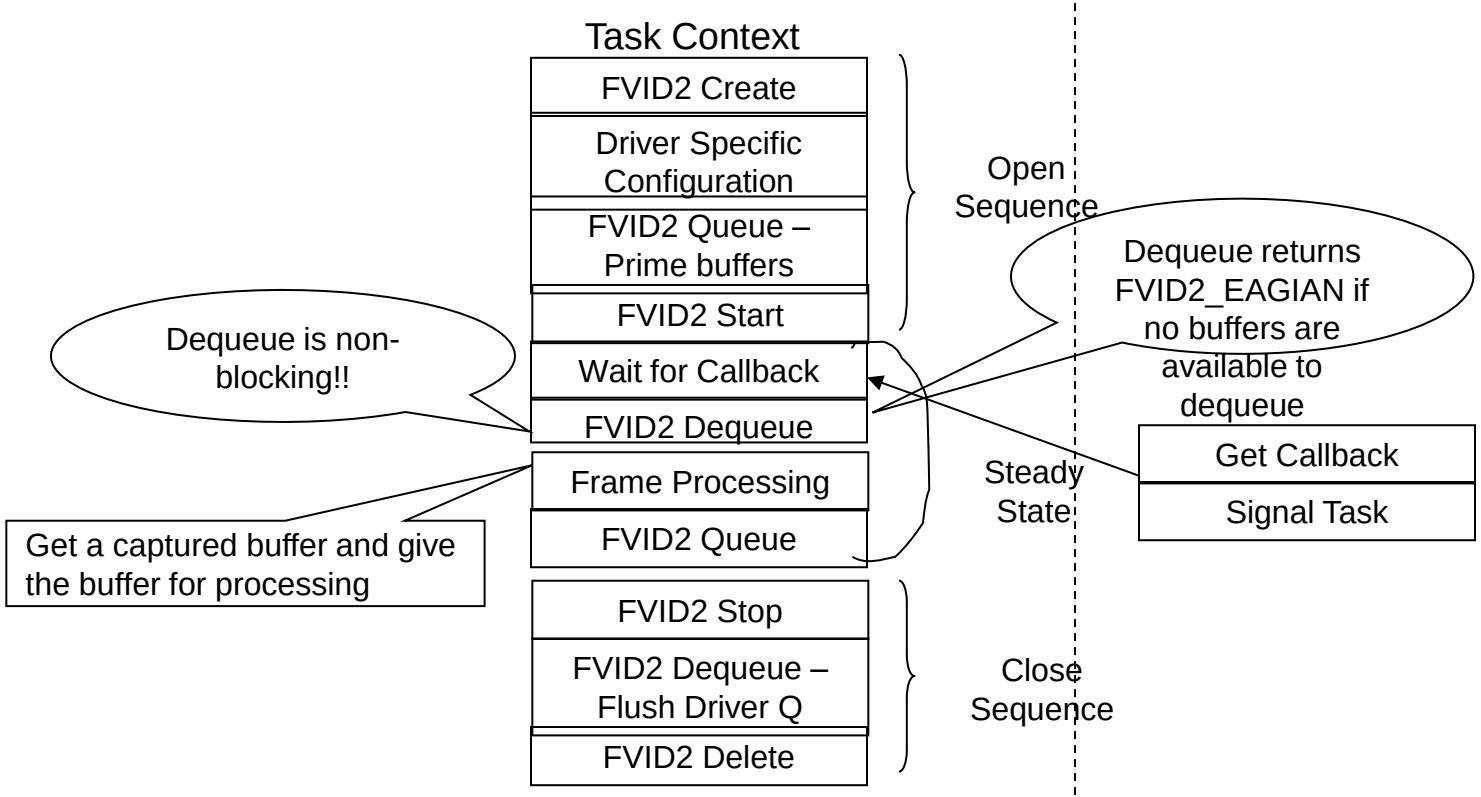


VIP Driver Overview (based on FVID2 Interface)

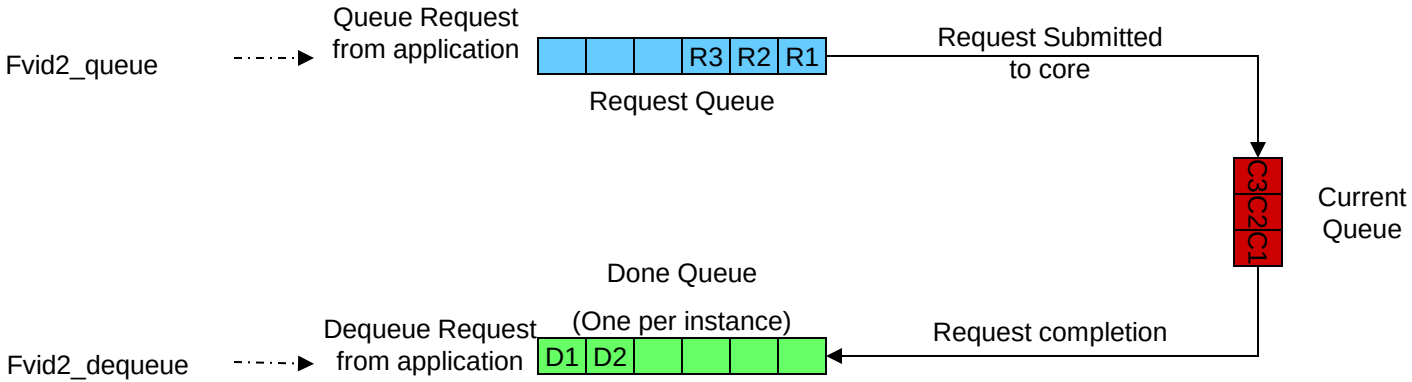
FVID2 Interface

FVID2_create	Init and prepare a VIP port for capture. Buffer allocation and priming must be done outside driver.
FVID2_start	Start the VIP port for capture
FVID2_dequeue	Get captured frames. Possible to dequeue all channels in a handle using single dequeue call. Need separate dequeue calls for each stream
FVID2_queue	After processing captured frames return them back to the driver. Also used for priming buffers to driver. Possible to queue from different channels in a handle using single queue call. Need separate queue call for different streams
FVID2_control	IOCTL like control to change frame drop sequence per channel, scaling ratio per stream, get channel status like captured width x height
FVID2_stop	Stop the VIP port
FVID2_delete	Release resources allocated during capture

Typical Application Flow



Queue Element Flow Diagram



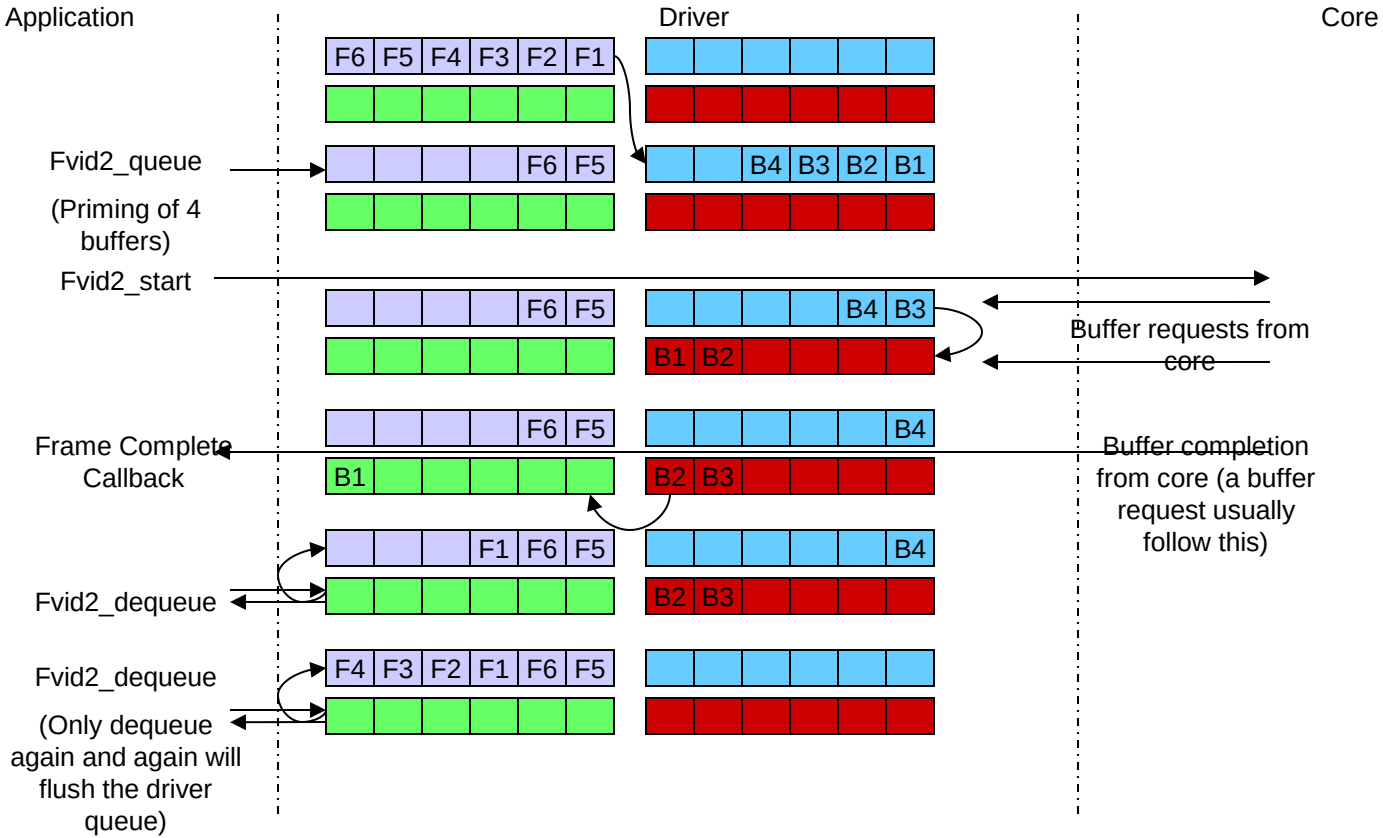
Buffer Capture Modes

- Frame Drop Mode
  - VIP Capture driver will stop capturing data when there are no more buffers at the input queue
  - It will not hold any buffer with it and the last buffer will be returned to the application through dequeue call
  - It makes use of the VPDMA drop data feature
- Last Frame Repeat Mode
  - VIP Capture driver will keep capturing the data to the last buffer when there are no more buffers at the input queue
  - It will hold the last buffer with it till the applications queues any new buffer or the capture is stopped
  - Used for on-the-fly capture to single OCMC-CBUF use case (without ping-pong buffers)

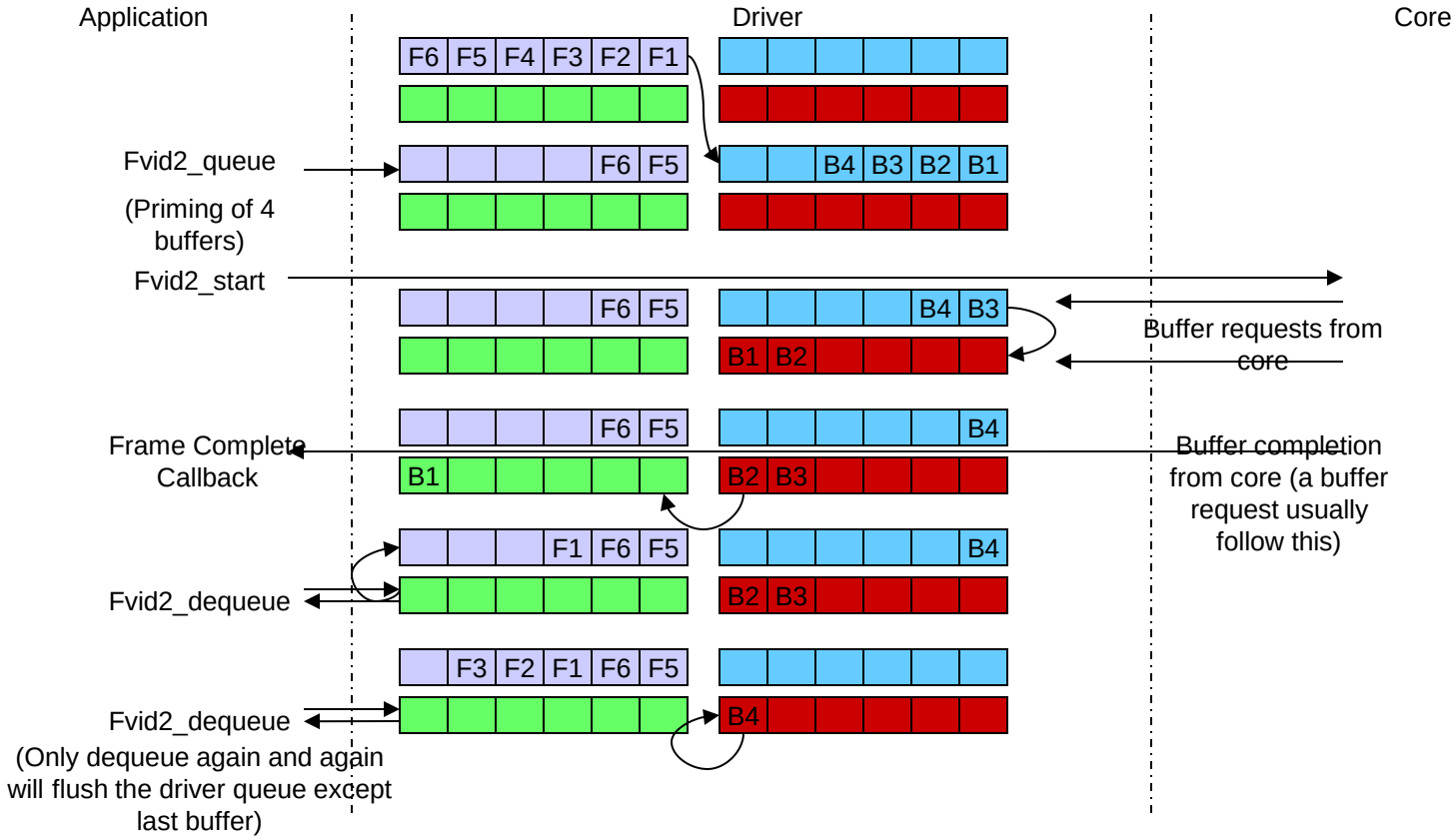
Contd..

- Circular Frame Repeat Mode
  - VIP Capture driver will keep reusing all the sets of buffer with it in a circular fashion
  - Application cannot get back any buffer from the driver when streaming is on and dequeue call will result in error
  - Used for on-the-fly capture to OCMC-CBUF use case with ping-pong buffers

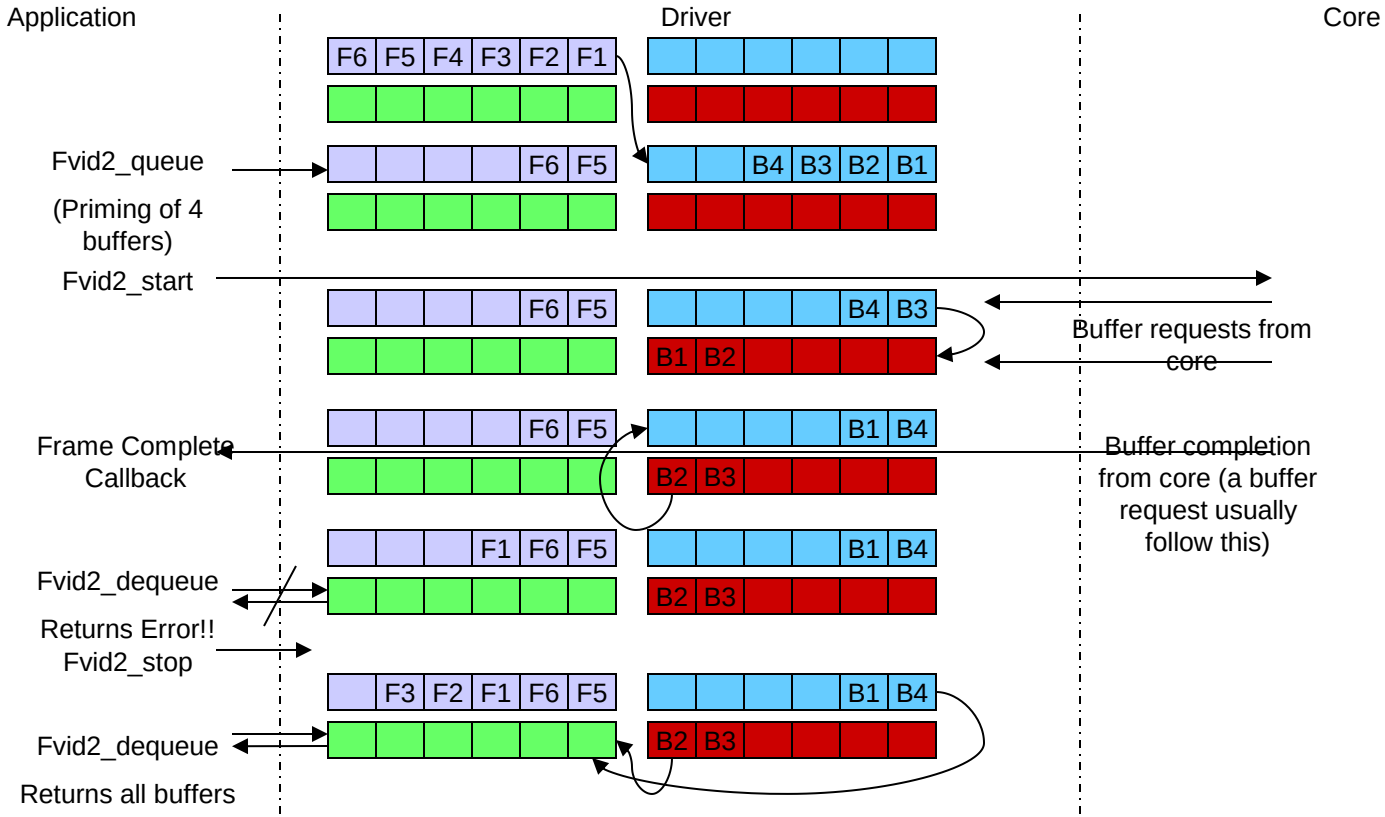
Frame Drop Mode Buffer Flow



Last Frame Repeat Mode Buffer Flow



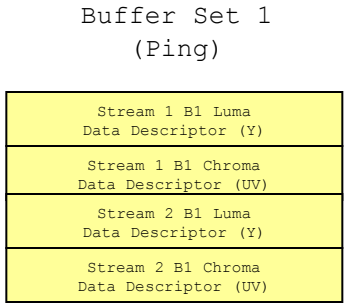
Circular Frame Repeat Mode Buffer Flow



VIP Driver List Management

Single Channel List Layout

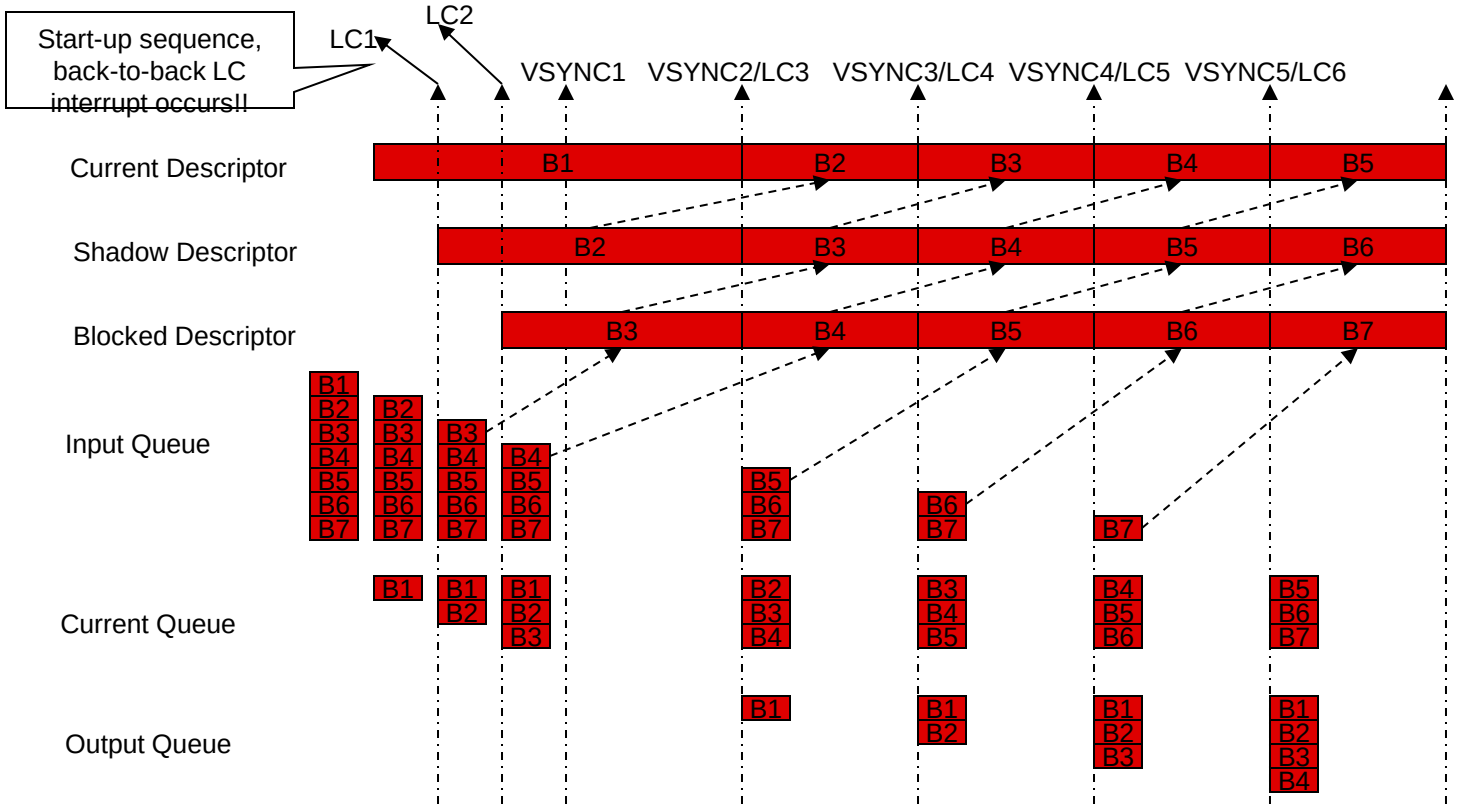
- Program descriptors every LC interrupt received
- Since post is based on VPDMA LC interrupt, the shadow descriptor gets filled in case of startup, short frames or emulation break points
- From 3rd LC interrupt, this could be considered as VSYNC



Contd..

- Use List Complete (LC) Interrupt
  - Acts like VSYNC interrupt since this happens after blocking descriptors are loaded except for the first two interrupts (which happens immediately)
  - Driver programs the next set of buffers and move the buffer programmed two LC interrupt earlier to output buffer
  - First two LC interrupts: Both descriptors gets accepted as shadow descriptor is not present for the startup

Single Channel: Timing Diagram



Questions?
Thank You