SM8250 Linux Android Camera Overview

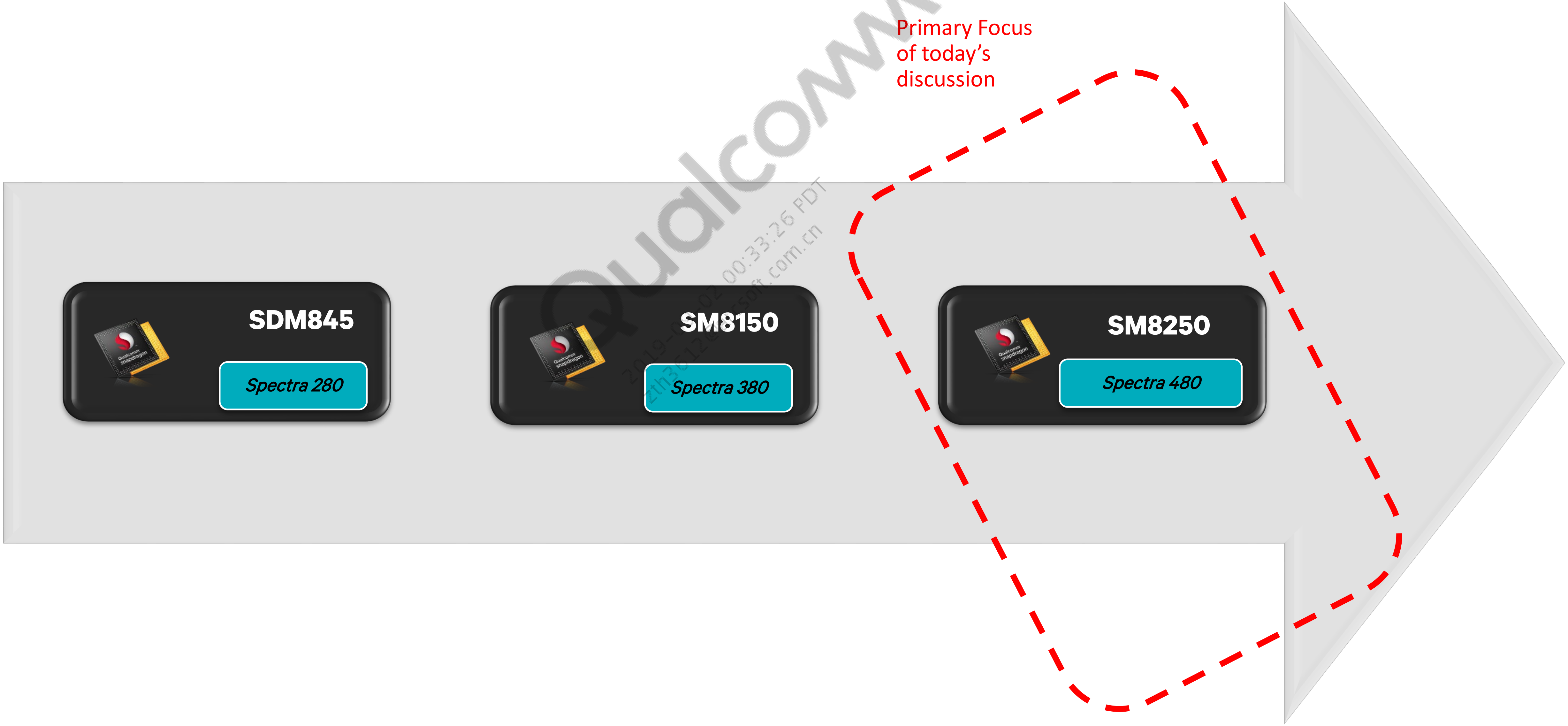
80-PK882-33 Rev. E

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision History

Revision	Date	Description
A	April 2019	Initial release
B	May 2019	Multiple editorial changes
C	May 2019	Added slides 21-30
D	June 2019	Updated slide 18
E	July 2019	Removed video super resolution (VSR)

Qualcomm ISP Roadmap for Snapdragon



SDM845

Spectra 280

SM8150

Spectra 380

SM8250

Spectra 480

Primary Focus of today’s discussion

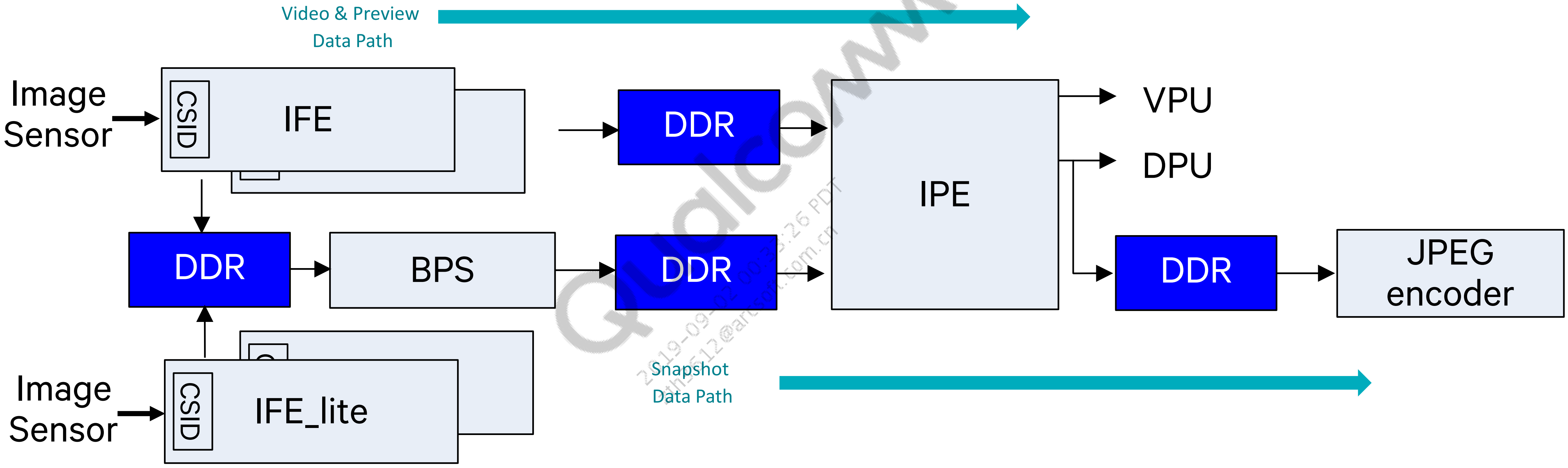
Qualcomm Spectra 380 vs. Qualcomm Spectra 480

	Spectra 380	Spectra 480
Maximum real-time sensor input resolution	2x Front-end input: - 22MP (4:3), 16MP (16:9 aspect ratio) 2x Front-end input : - mono/YUV Interface for misc. CV use-cases.	2x Front-end input: - 25MP (4:3), 18MP (16:9 aspect ratio) 2x Front-end input : - mono/YUV Interface for misc. CV use-cases.
Camera interfaces	DPHY 1.2: 4/4/4/4 (2.5 Gbps/lane) CPHY 1.0: 5.71 Gbps/T Connect up to 8x cameras, 4x concurrent	DPHY 1.2: 4/4/4/4/4/4 (2.5 Gbps/lane) CPHY 1.2: 10.28 Gbps/T Connect up to 12x camera, 4x concurrent
ZSL performance	2x 20M@30fps – dual camera 1x 32M@30fps – single camera	2x 25M@30fps – dual camera 1x 64M@30fps – single camera
Video	10-bit 4K/60 Support with in-line hardware EIS 1080P@240/720P@480	10-bit 4K/120, 8K/30 Support with in-line hardware EIS 1080P@480 / 720P@960
IQ and imaging improvements	**Enhancements over Spectra 280** • Face Detection HW improvement • zzHDR improvement **New for Spectra 380:** • 2PD HW support	**Enhancement over Spectra 380** • PDPC for R, B and G. 2x1 and 2x2 OCL • Adaptive lens shading correction to recover highlight details (LSC) • PDLib for 2PD and sparse PD • ML based face detection • zzHDR IQ improvement on jagged lines and motion artifacts • MCTF with local motion compensation **New for Spectra 480** • Binning correction to remove artifacts due to sensor binning • Mid/low freq denoising improvements and local contrast enhancement (LENR)
Tap-out points	• YUV 4:2:0 • Bayer • RGB	• YUV 4:2:0 • Bayer • RGB

Qualcomm Spectra 380 vs. Qualcomm Spectra 480 (cont.)

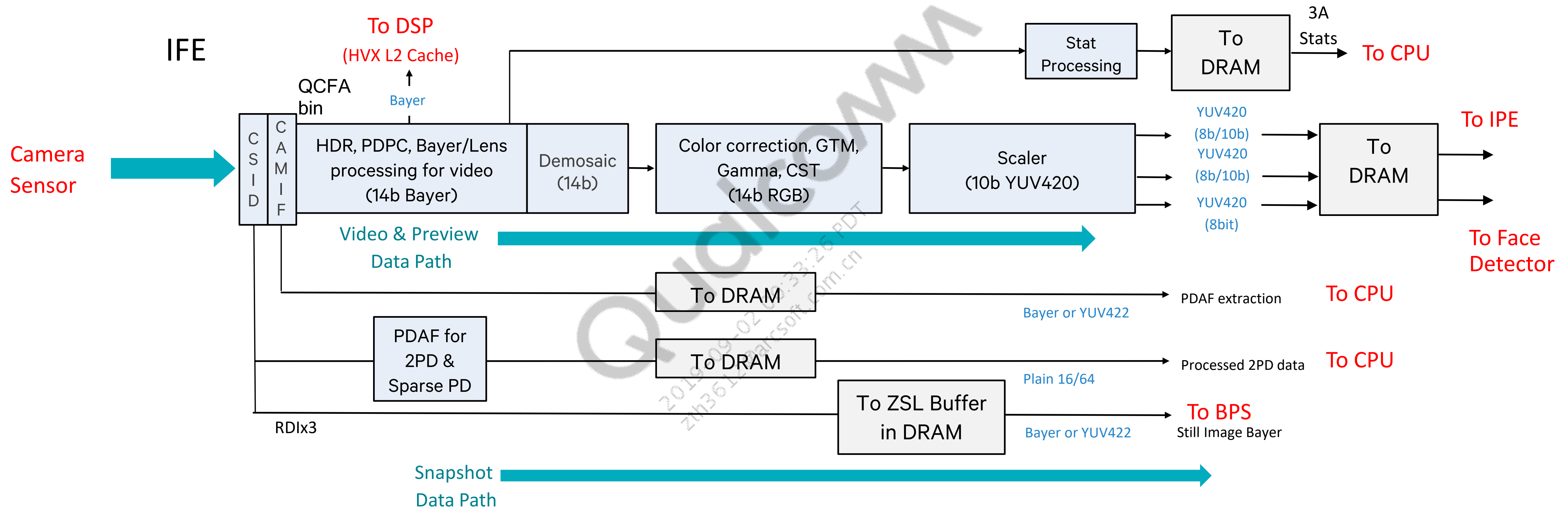
	Spectra 380	Spectra 480
Motion estimation for video	LRME in ISP	CVP
Power improvements	Power enhancements in Spectra 380 • Optimization to achieve higher frequencies • Separate power rail • EIS processing fully within camera subsystem without the need of GPU. • UBWC 3.0	Power enhancements in Spectra 480 • Same use case at lower voltage • Better compression to reduce BW and power consumption • 4K@120fps without thermal impact • UBWC 4.0

High Level Architecture Overview of Qualcomm Spectra 480



- **CSID**: Camera Serial Interface Decoder Module
- **IFE** (x 2): Image Front End
- **IFE_Lite** (x 2): Image Front End Lite
- **BPS**: Bayer Processing Segment (for snapshot)
- **IPE**: Image Processing Engine
- **VPU**: Video Processing Unit (CODEC)
- **DPU**: Display Processing Unit

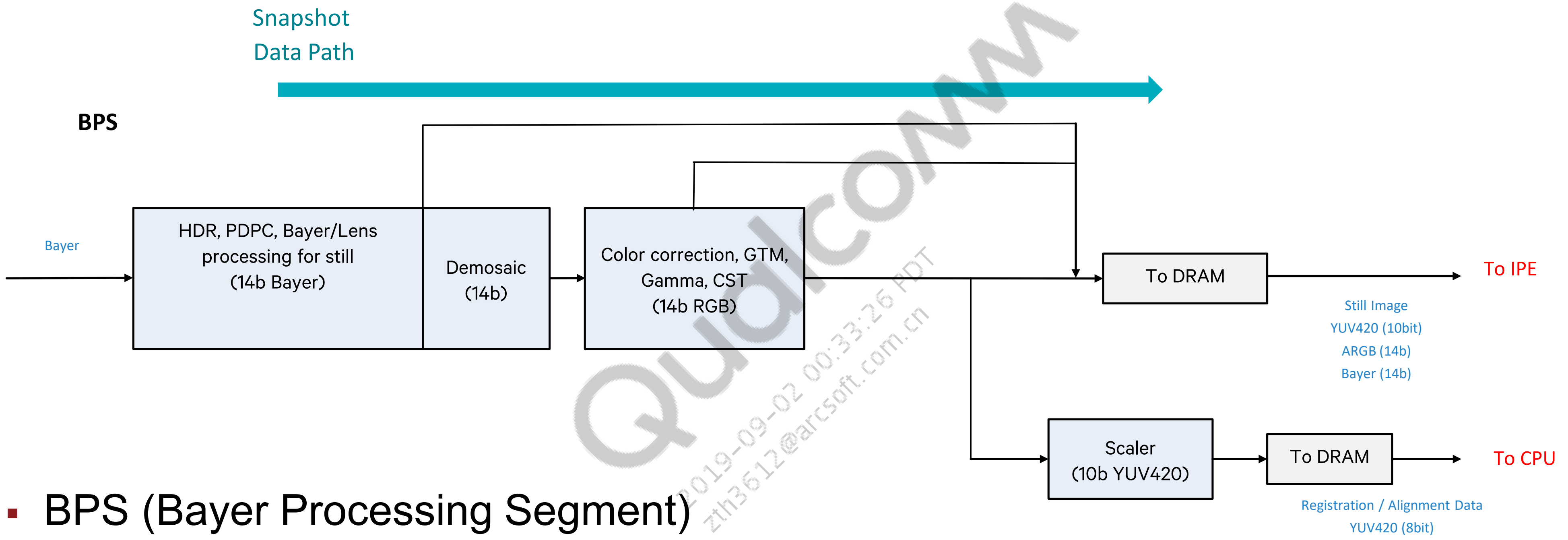
High Level Architecture Overview of Spectra 480: IFE



- **IFE (Image Front End)** x 2
  - 12MP@120 each
  - Bayer processing for video/preview
  - Stats for 3A (Bayer grid/histogram, Bayer focus, etc.)
  - 4-tap-in/out points for HVX streaming
  - PD correction (including 2x1/2x2 OCL), zzHDR, QCFA binning
  - PDLib for 2PD and sparse PD (AF improvements)
  - Separate preview and video output

- **CSID**
  - CSI Decoder
  - 3/4 RDI (Raw Data Interface) output to DDR
  - IFE_lite is separate from IFE (with 4 RDIs)

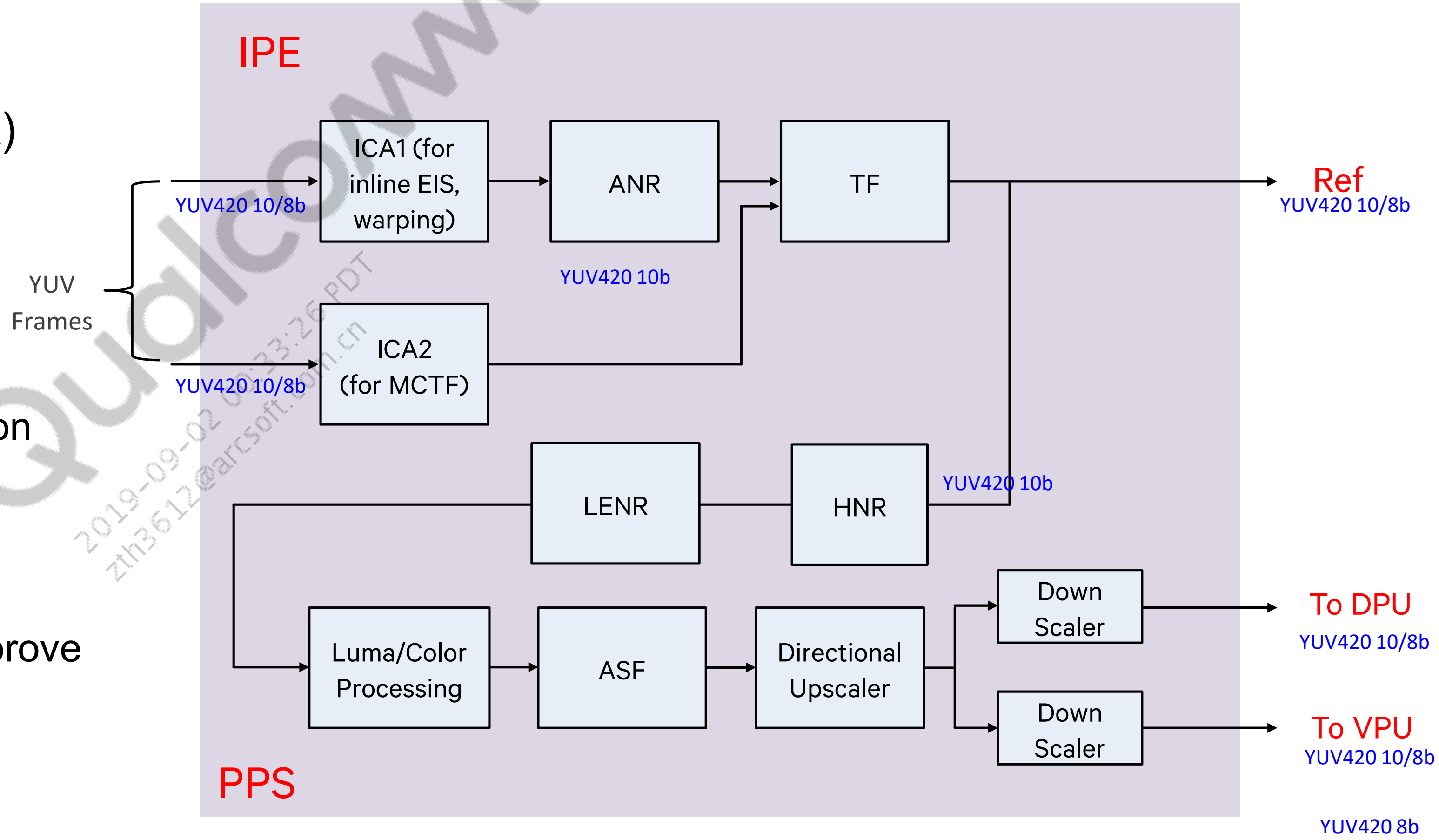
High Level Architecture Overview of Spectra 480: BPS



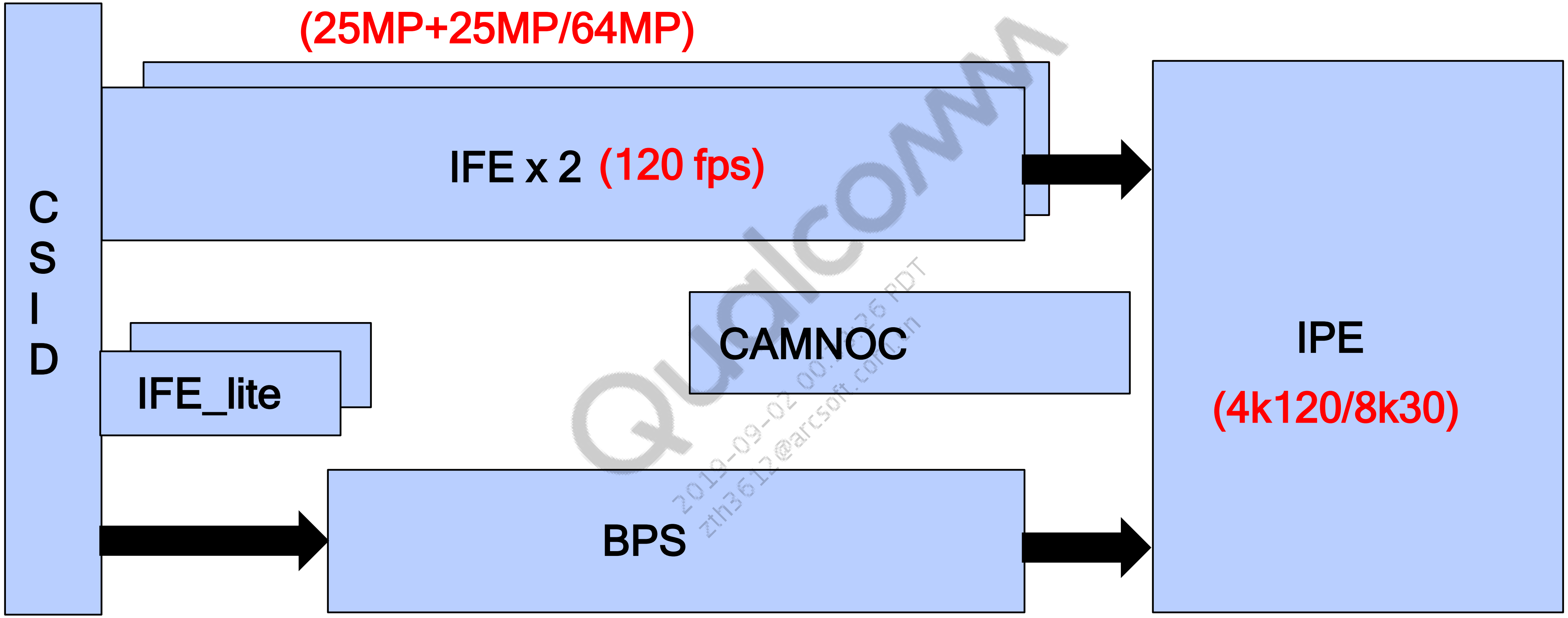
- BPS (Bayer Processing Segment)
  - Snapshot path only
  - Bayer processing (e.g., demosaic, zzHDR, PDAF pixel correction (including 2x1/2x2 OCL), lens shading correction, etc.)
  - Downscaling for registration and alignment of sequential frames for multi-frame processing
  - Multiple format support for flexible hybrid of HW/SW
  - No HNR in BPS

High Level Architecture Overview of Spectra 480: IPE

- IPE (Image Processing Engine)
  - 4k120/8k30 performance
  - ICA (Image Correction and Adjustment)
    - Warping engine (LDC, inline EIS)
  - ANR (Advanced Noise Reduction)
    - Multi-pass processing
  - TF (Temporal filtering)
    - Temporal noise reduction with local motion compensation for video
    - Multi-frame processing for snapshot
  - HNR
    - Transform domain noise reduction to improve high frequency noise
  - LENR
    - Low-middle frequency noise reduction
  - LTM, color processing
  - ASF (Advanced Spatial Filtering)
  - Directional upscaling



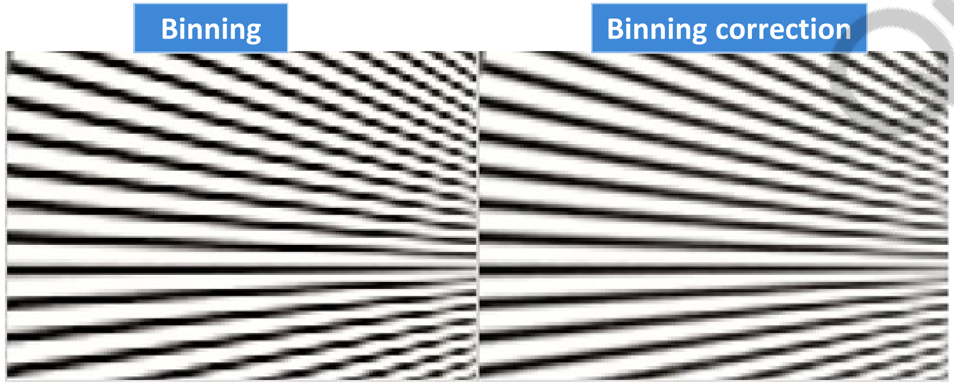
Enable High Performance Use Case within Thermal Limit

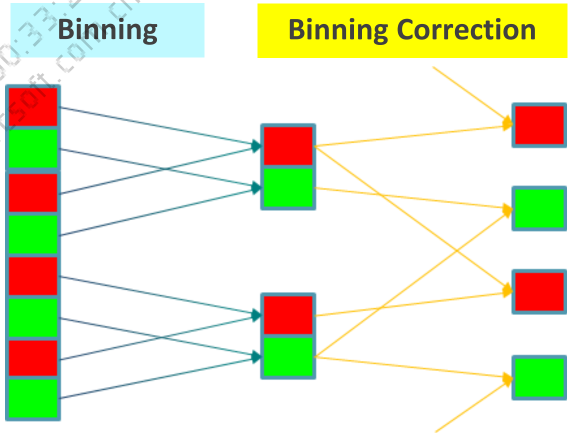


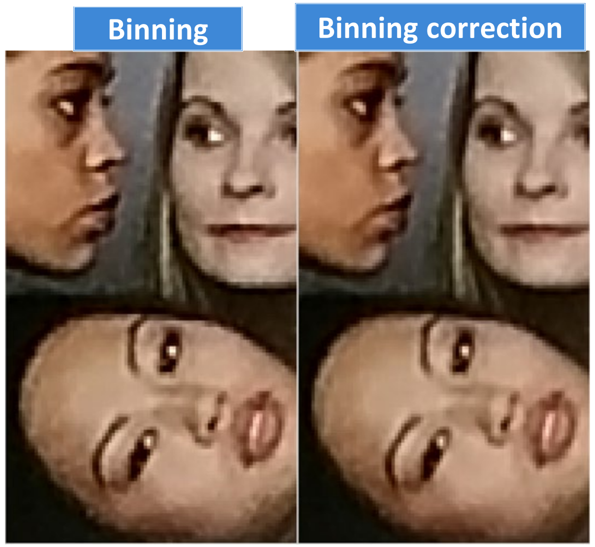
- New architecture for high performance use case
  - Enable 4k120, 8k30, and 720p960
  - Enable high-read-out sensor 12MP@120 with single IFE (dual 12MP@120 with 2 IFEs)
  - 20%+ power savings from Spectra 380 for UHD60
- 25MP+25MP/64MP sensor support
- 4 concurrent cameras with 2 IFE and 2 IFE_lites

Bayer Binning Correction

- To support sensor binning for HFR, and power savings
- Sensor binning generates zig-zag artifacts because of wrong phase
- Binning correction corrects the phases and smooth edges







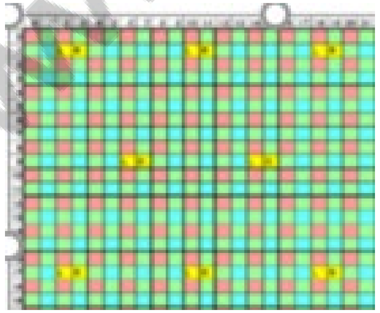
PDAF – PD Pixel Correction

PDPC:

PD type	Spectra 280	Spectra 380	Spectra 480
Sparse PD – R, B ch	Y	Y	Y
2x1/2x2 OCL	N	N	Y
PD+zzHDR	N	N	Y

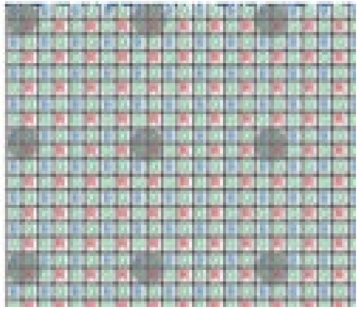
- Spectra 280/380
  - Correction on R and B ch (no G ch correction)
  - Need to use strong noise reduction for PD on G ch (e.g., 2x1) => quality degradation
- Spectra 480
  - Added G ch correction (good for 2x1/2x2 OCL)
  - No need to trade IQ for stronger noise reduction
  - PD+zzHDR: only using green pixels from the same exposure for gradient

2x1 OCL

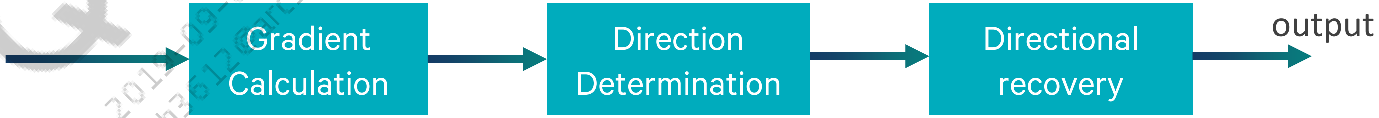


PD pixel on 2x1 pixel pair (including G ch)

2x2 OCL



PD pixel on 2x2 pixel pair (including G ch)



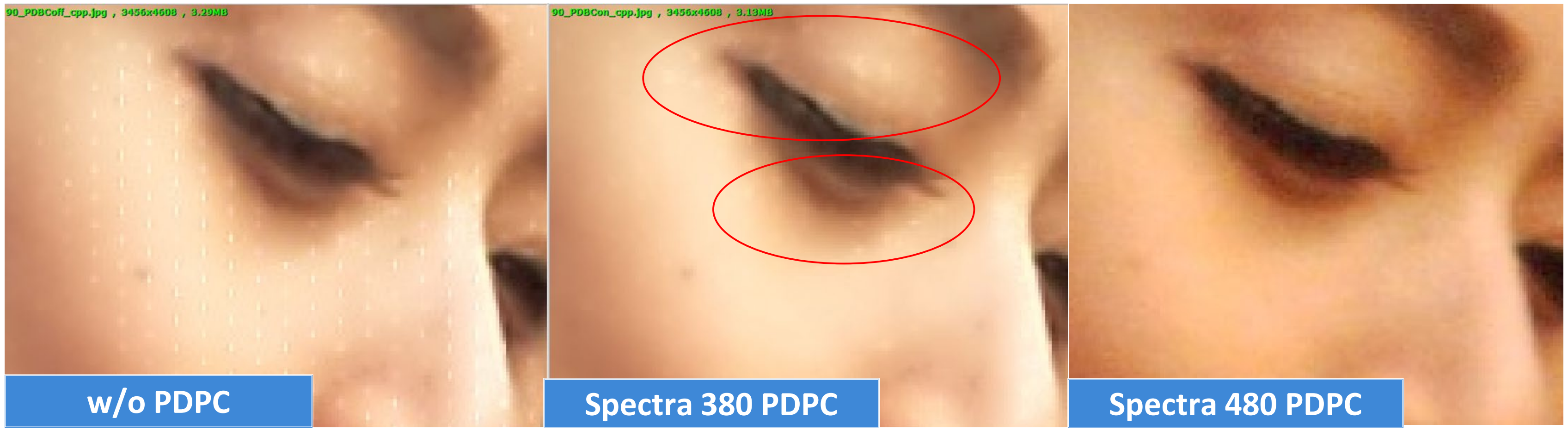
Gradient Calculation

Direction Determination

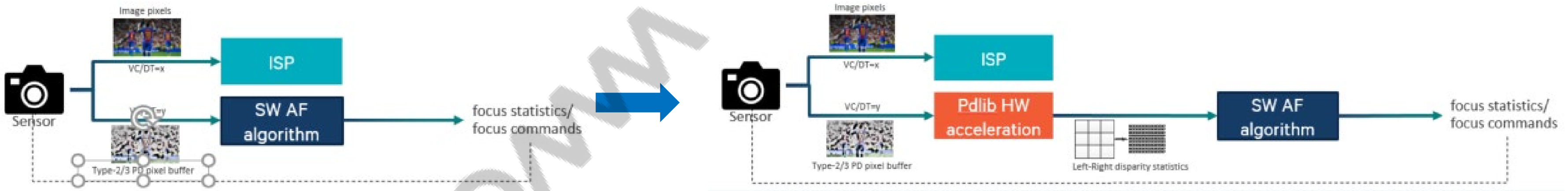
Directional recovery

output

Need to use stronger NR => affecting IQ


90_PDBCoff_cpp.jpg , 3456x4608 , 3.29MB
90_PDBCon_cpp.jpg , 3456x4608 , 3.13MB
w/o PDPC
Spectra 380 PDPC
Spectra 480 PDPC

PDAF – PD Lib

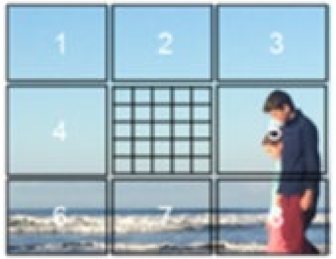


PDLIB HW:

- Spectra 280
  - No HW PDLIB (Used SW)
  - Latency and power are too high for full frame focus
- Spectra 380
  - HW PDLIB for 2PD
  - Offload PD data processing from CPU
  - Reasonable latency and power for full frame focus
- Spectra 480
  - HW PDLIB for 2PD and sparse PD/2x1 OCL

PD type	Spectra 280	Spectra 380	Spectra 480
2PD	N	Y	Y
Sparse PD	N	N	Y
2x1/2x2 OCL	N	N	Y

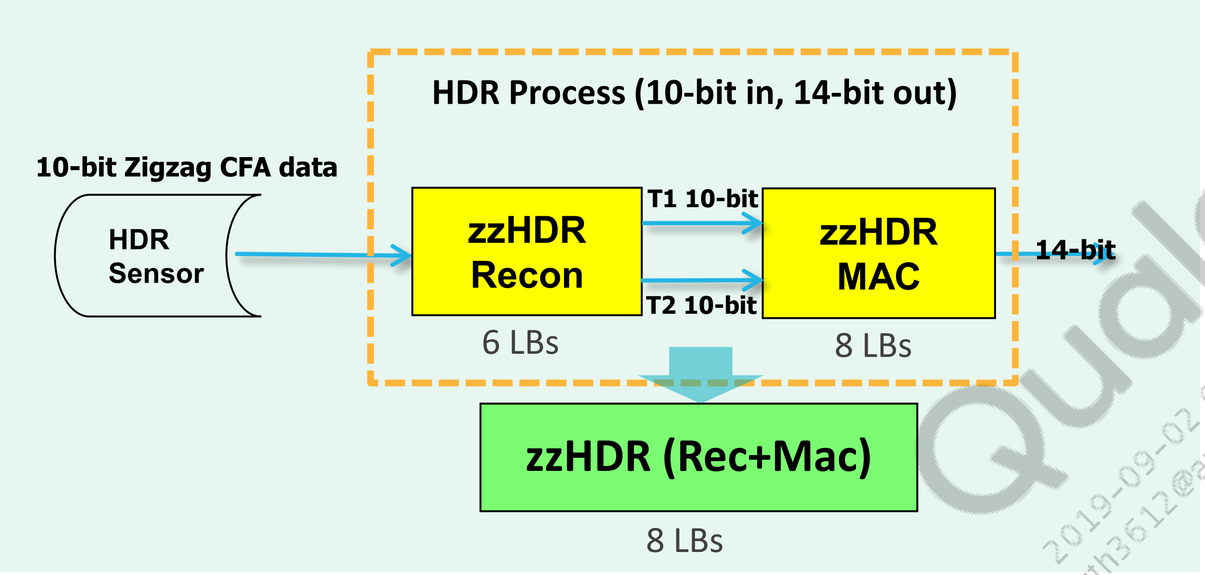
Full Frame Focus

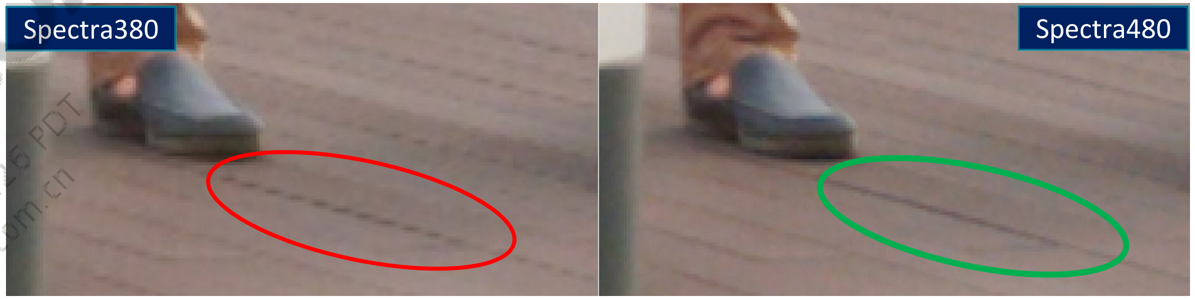


Projected performance/power for Full Frame Focus

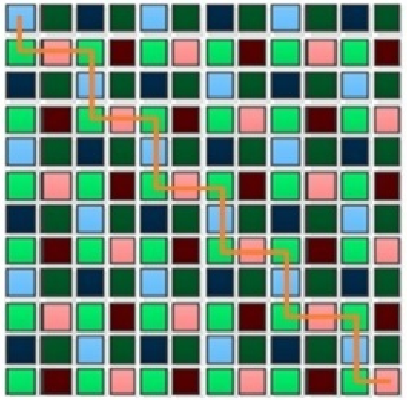
- Benefits:
  - AF performance improvements: PDAF HW enables 100% FOV with full quality
    - SW only cannot achieve 100% FOV with full quality (typically 30% FOV)
  - AF latency improvements
  - Power: CPU power reduction

zzHDR Improvements





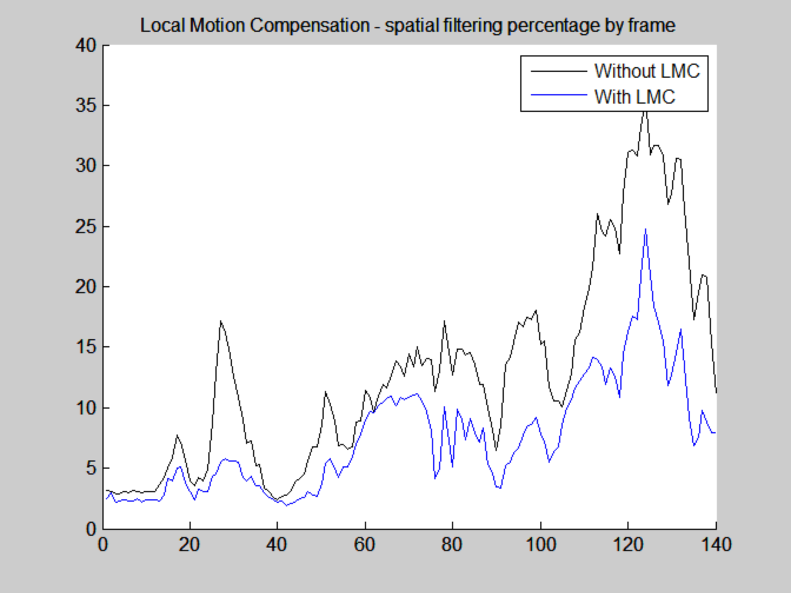
- Quality Improvements
  - Improved T1/T2 interpolation
  - Compensates errors on exposure ratio
  - Better motion detection => improved exposure ratio for better dynamic range



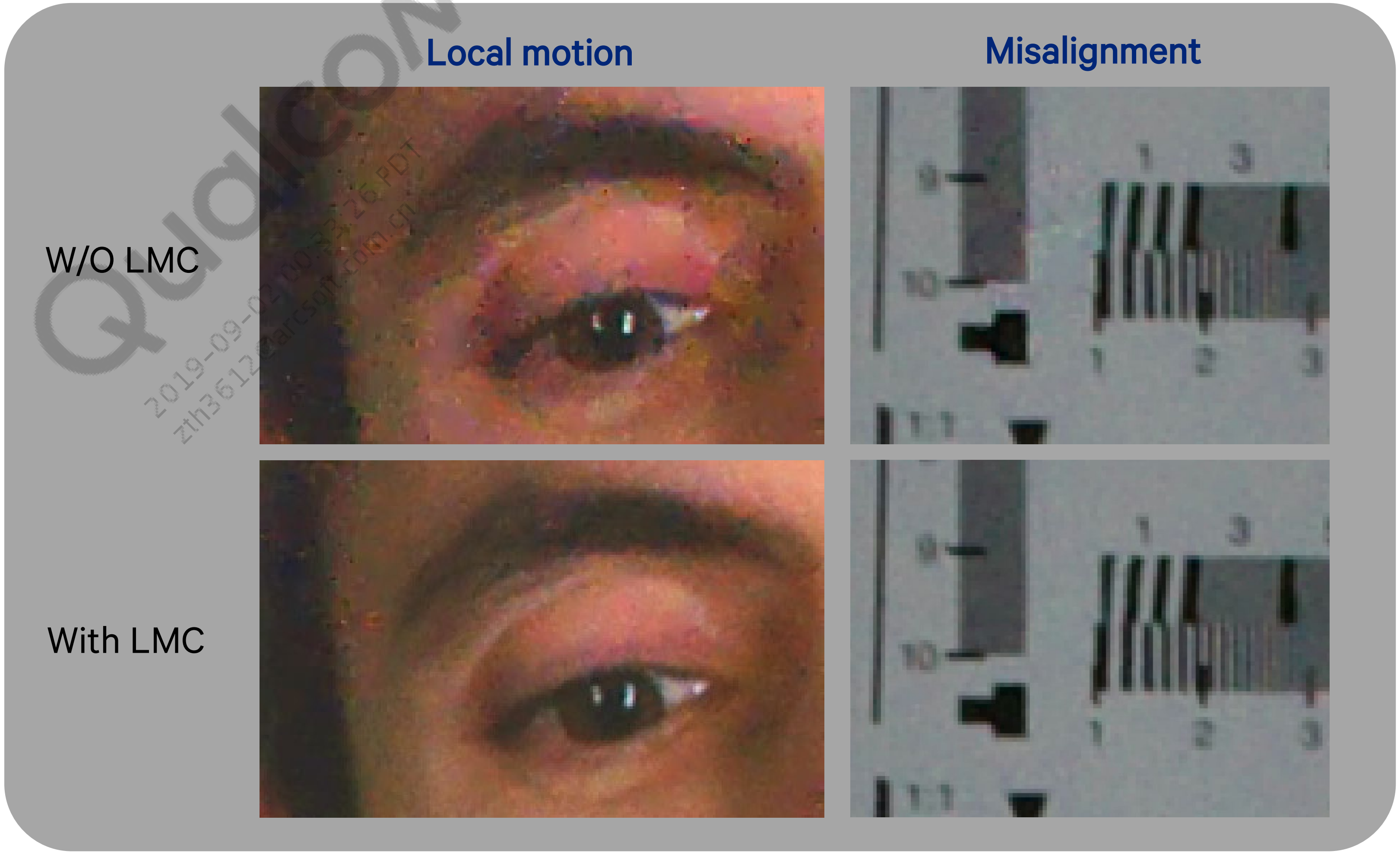
zzHDR Pattern

MCTF Improvements – Local Motion Compensation

- Spectra 280/380
  - Global motion compensation
  - No temporal filtering for local motion or non-accurate alignment
- Spectra 480
  - Global and local motion compensation

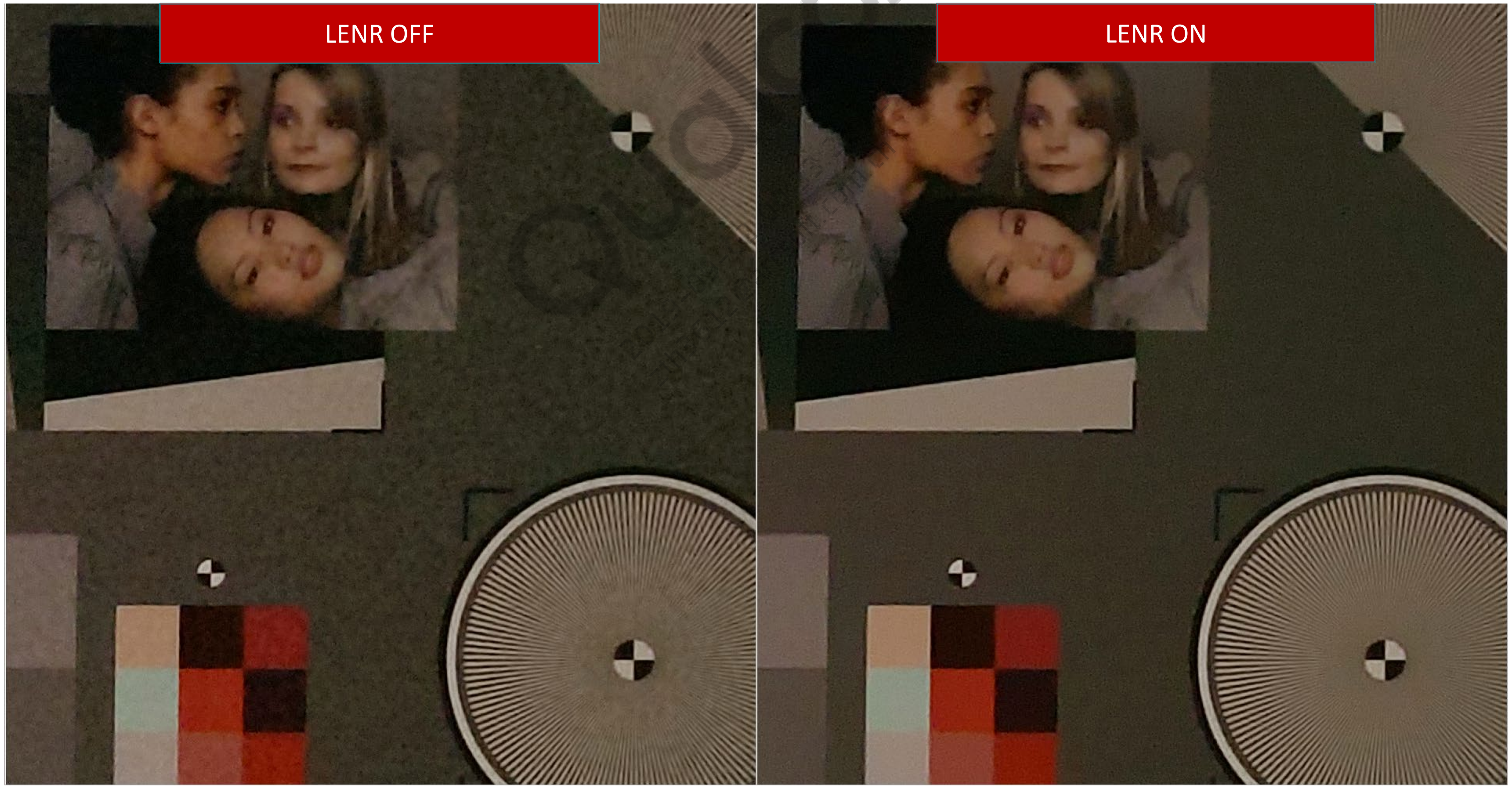


TF rate (higher is better)

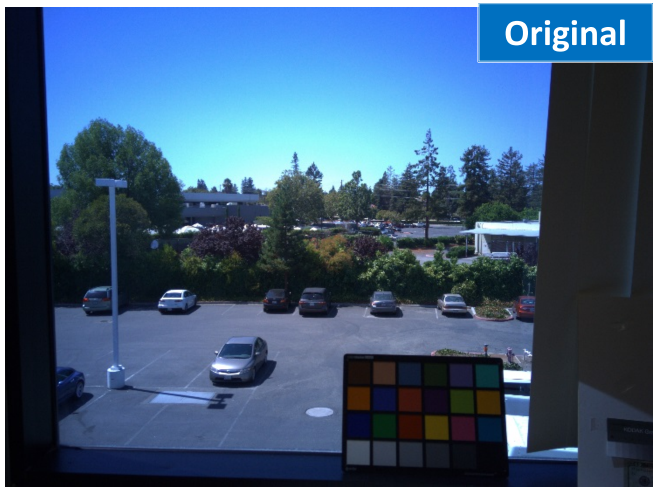


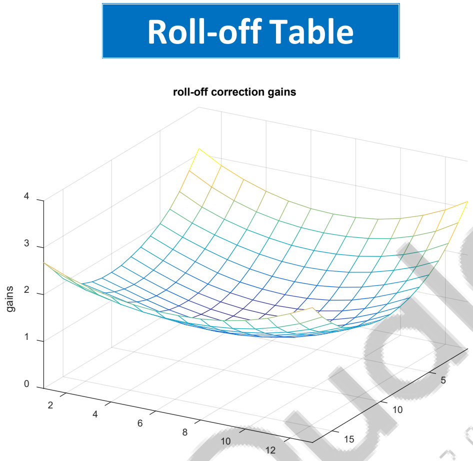
LENR – Low/Mid Frequency Enhancement and Noise Reduction

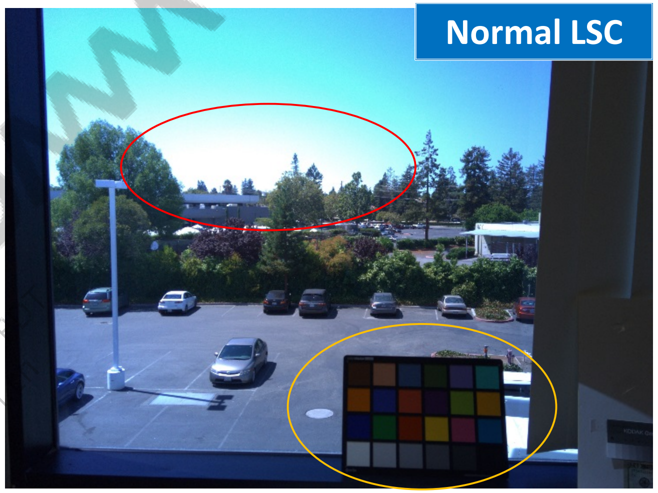
- Purpose of LENR
  - To reduce mid/low frequency luminance noise and enhance local contrast.
- LENR is designed to keep more details while denoising.

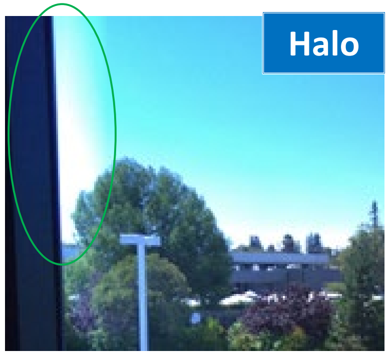


Adaptive Lens Shading Correction – Keeping Highlight Detail

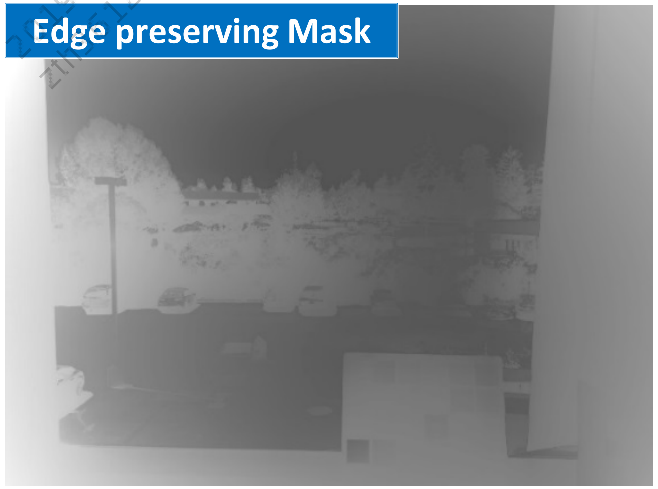

Original

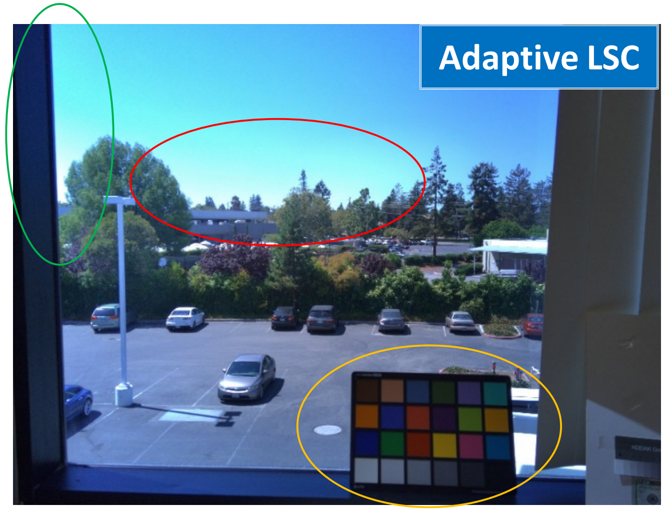

Roll-off Table


Normal LSC


Halo

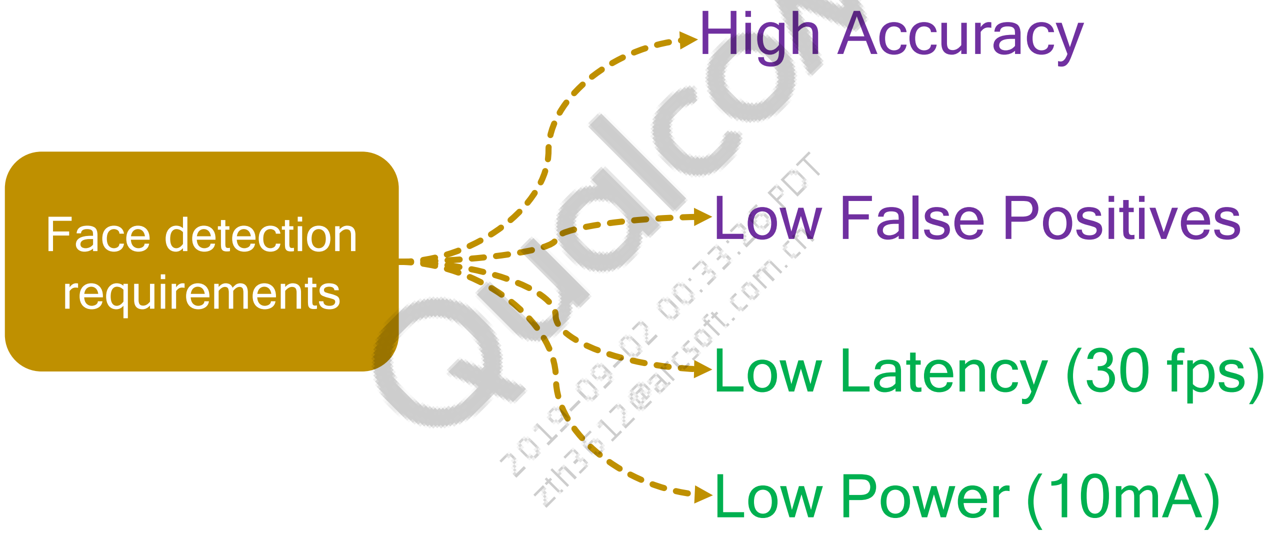
- Recover the highlight details that cannot be recovered later in the pipeline
- Serves as simple Local Tone Mapping
- Edge-preserving mask to avoid halo artifacts


Edge preserving Mask

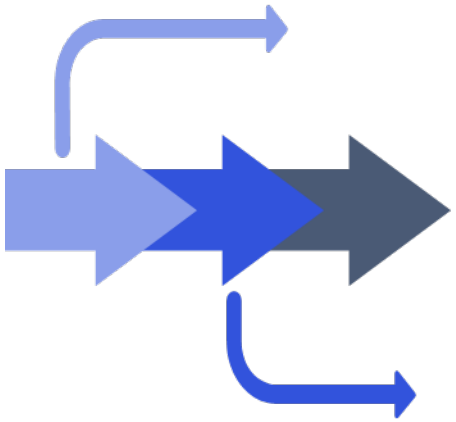

Adaptive LSC

Face Detection based on Machine Learning

- Continued improvements on detection rate, false positives, small face, and distance



ISP	Spectra 380 FD 5.0	Spectra 480 FD 6.0
Detection rate	73%	82%
Smallest face	25x25	25x25
Distance	9ft	9ft
False positive filter	Conventional	Deep learning



SM8250 CHI Enhancements

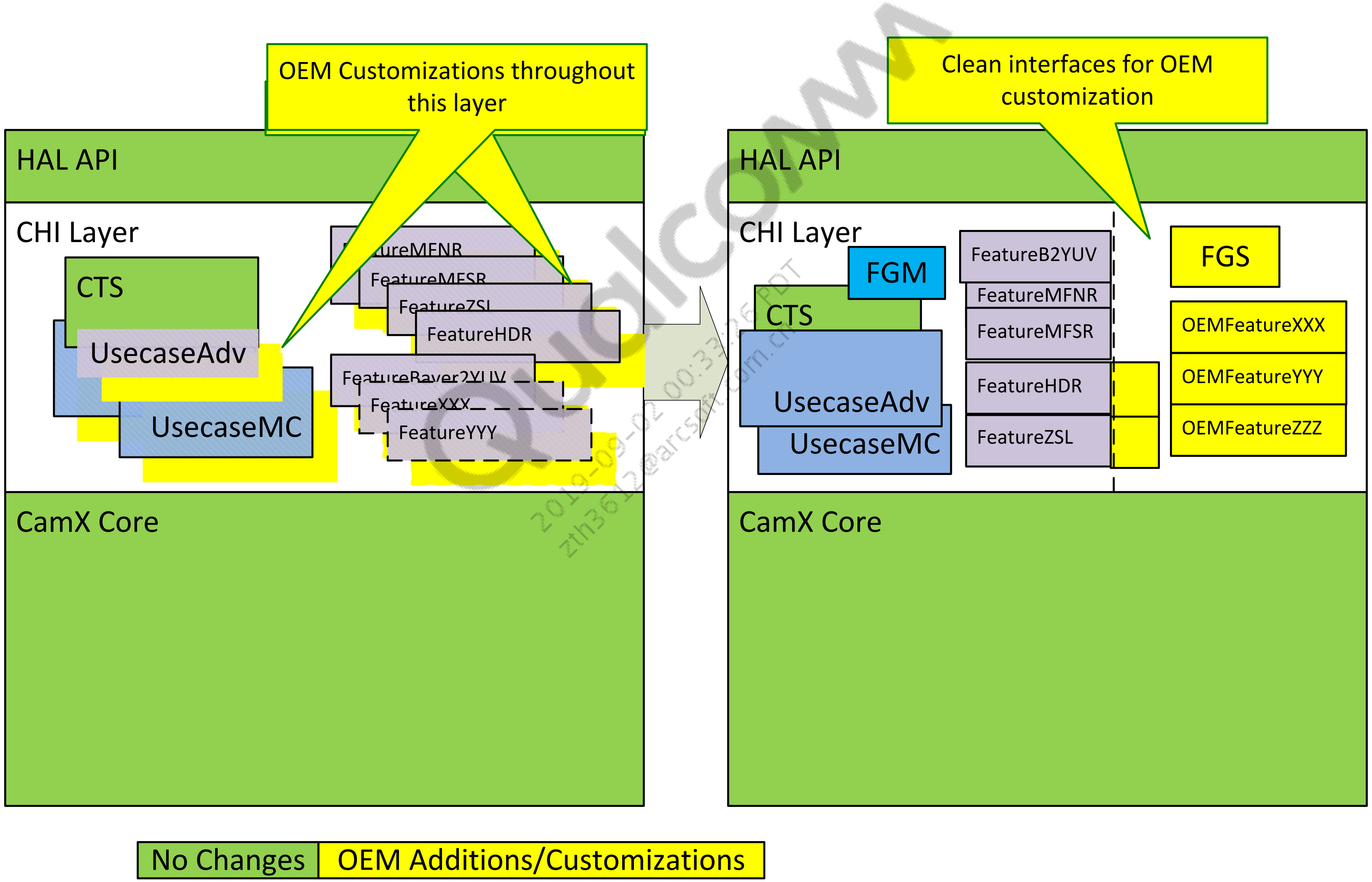
SM8150 CHI Scope for Improvements

Issue	Analysis
Large portion of issues introduced from CHI side in metadata/buffer management, flush/error handling, managing ZSL logic, etc.	Features are complex, as they: • Manage requests and their states • Have tight coupling with usecase • Handle metadata/buffer and their ref counting, for feature and non-feature related processing (e.g., realtime, JPEG, etc.) • Process result/partial-results • Handle ZSL logic • Are tightly coupled to realtime (sensor-based input) and can not easily support offline/background snapshot request processing • Often enable workarounds to keep realtime pipeline functional in case of delays within feature processing • Have duplication of such code across each usecase, often maintained by different developers per usecase. For each implementation, they also need to handle flush and errors reported from camx side (from a feature perspective) Most of this redundant work can be abstracted from customer by supporting a common base class for the feature.
Feature combination (e.g., MFNR+HDR) enablement was difficult (not scalable and error-prone).	Features are not modularized enough and need usecase-specific handling (e.g., single versus multi-camera need different handling usecases), managing state of requests, etc. Enabling feature combinations required a single feature to manage processing for both features and their requests.
Upgrading CHI between releases is time consuming.	Lack of clean separation of QCOM code versus customer customization.

SM8250 CHI Enhancement Goals

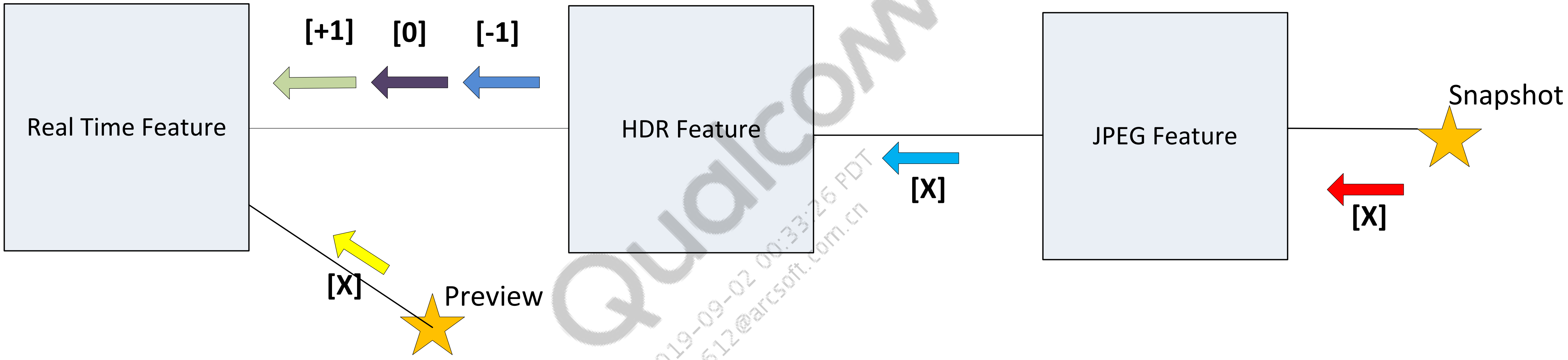
- Ease of adding features and supporting feature combinations (similar to how a custom node works)
  - Customers can focus on specifying what inputs/outputs they need for a feature
  - Customers do not have to worry about how to get/manage the input/outputs
  - Customers can unit test their derived feature separately without needing the whole usecase to work
  - Customers can easily “chain” features into a topology graph without need to manage setup or data/buffer/control movement within the graph
- Reduced integration time for post-CS releases, with better code modularization, enabling the customer to maintain only the customization code rather than common base class functionality
- Customers can leverage Qualcomm-tested baseline for common functionality, therefore focusing their development and stabilization efforts on their customizations

SM8250 CHI Architecture Enhancement



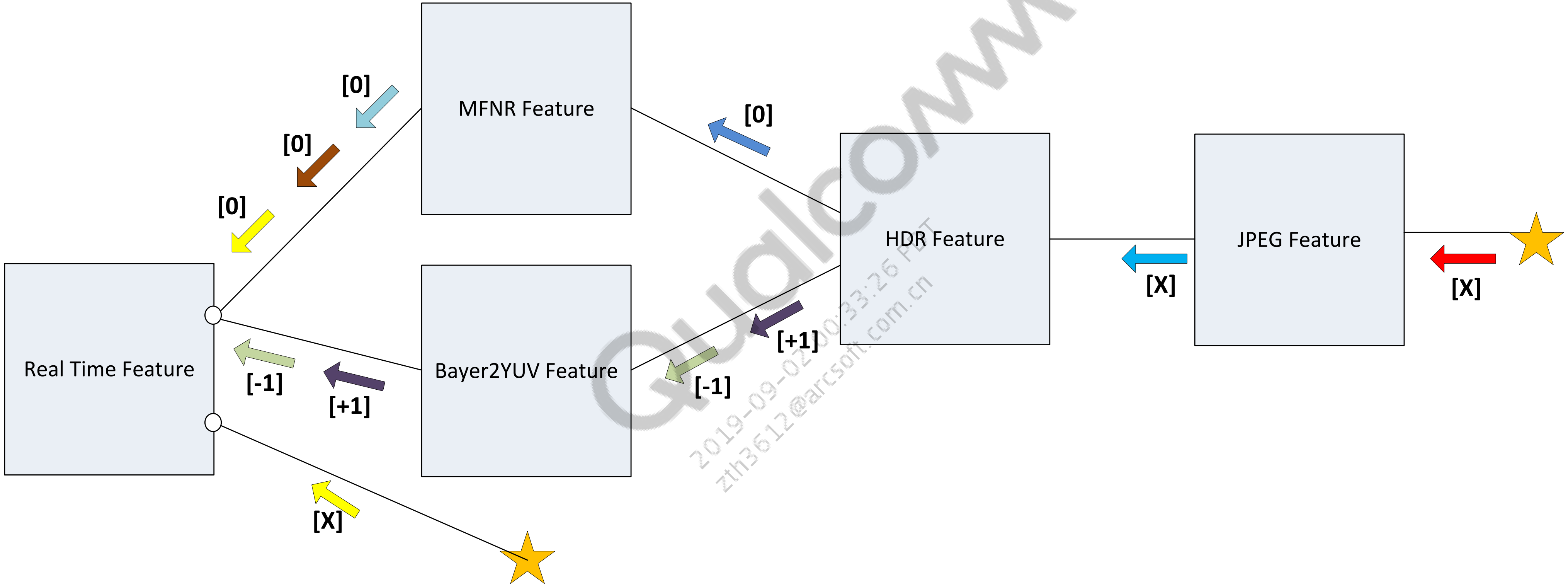
- *Left = Legacy CHI Architecture*
- *Right = SM8250 CHI Architecture*
- *FGM = Feature Graph Manager*
- *FGS = Feature Graph Selector*

Feature Graphs Example



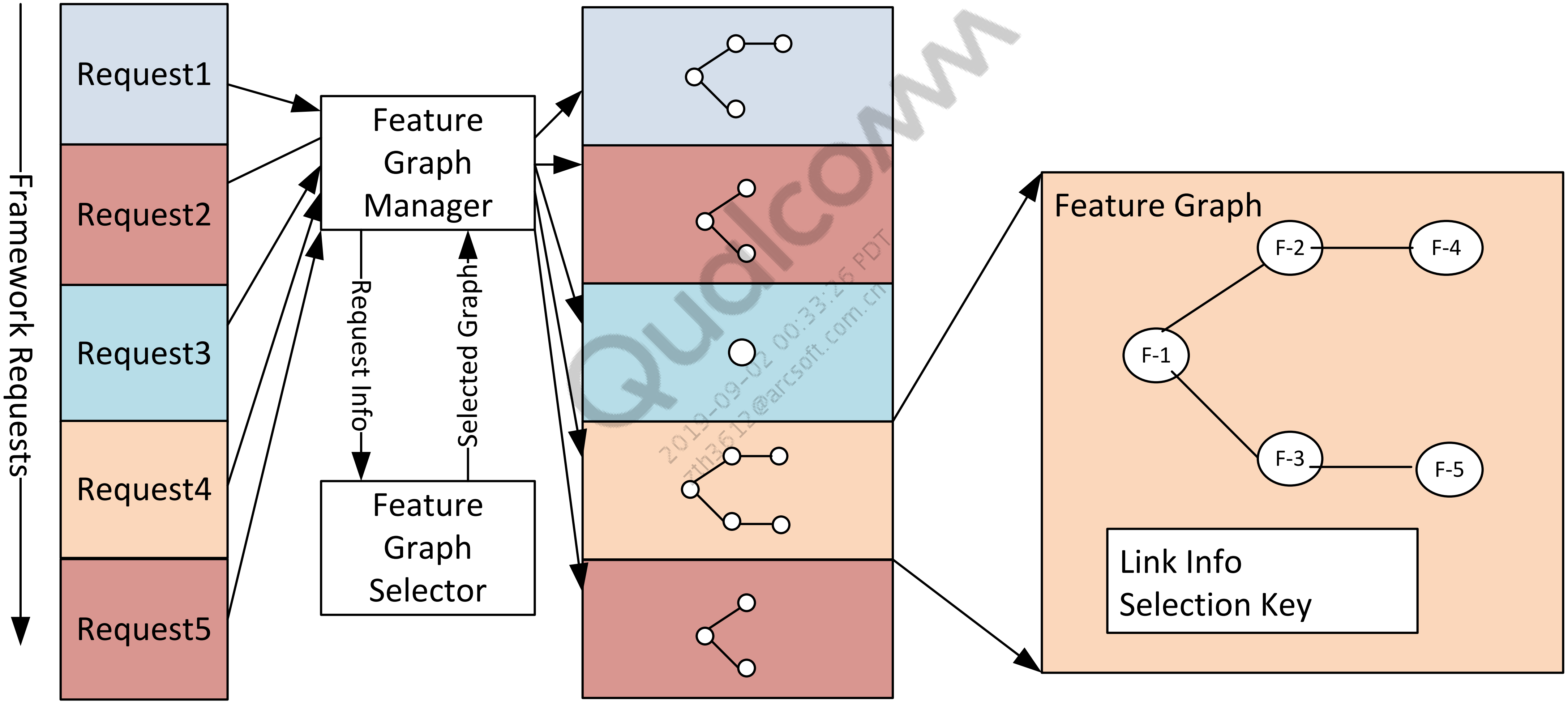
- Easy to specify a graph composed of features
  - Concept is similar to the existing CamX pipelines that are composed of a graph of CamX nodes
- Features can be developed, tested, and deployed completely independent of each other
- Customers can customize or build new graphs based on their needs

Advanced Feature Graph Example



- Dual camera usecase with MFNR+HDR (with common anchor) for snapshot

Feature Graph Selection



- Per-request graph selection based on the scene or other application indicated triggers
- Each request may use a different graph
- One or more graphs may be concurrently active depending on the nature of the processing in each graph

Comparison with the Legacy Feature

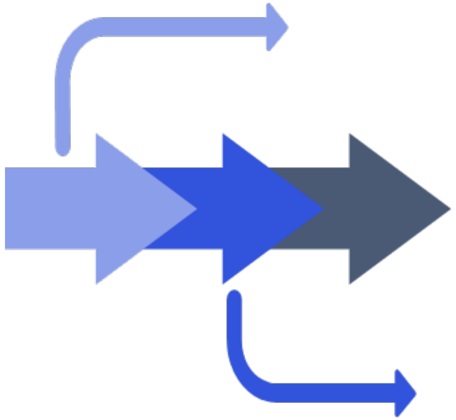
Stage	SM8150 Feature	SM8250 Feature2
Creation	• Create feature derived class • Add custom stream, meta, buffer handling, fence management • Manage stream-to-buffer mapping for every stream in every supported use case • Manage RDI snapshot buffer picking logic • As needed, add thread management logic • Differentiate between live/ZSL buffers • Manage and schedule buffer movement between pipelines • Manage sequencing of flush operations for each buffer manager and pipeline	• Create feature2 derived class • Add description of feature's inputs, outputs, dependencies, and stage in table format
Customization	• Add feature-specific logic within the feature code	• Add feature-specific logic within the context of customization hooks
Integration	• Update code within all usecases to utilize the feature • For advanced usecases, add code to manage realtime requests • For usecasemc, modify to manage inputs and outputs for this new feature	• Update feature graph selector (FGS) to make use of the feature
Feature Testing	• Unit testing is not possible • Customers do end-to-end functional testing for each usecase	• Unit test the feature with the provided test harness

Rollout Timeline

- Implementation to be completed by 8250 FC
  - Changes will be gradually added over ES1 and ES2 (but activated in FC)
- The goal to meet all the CS metrics
- Incremental documentation and customer trainings through ES1, ES2, and FC

Requested Actions for Customers

- Confirm if any changes required in feature2 base class or interfaces are needed to support customer needs. It is expected that customers will not add any customization within the feature2 base class.
- Share details of any customizations made in SM8150 on Qualcomm delivered features (e.g. ZSL, MFNR, HDR, etc.), so that we can enable clean hooks for enabling such customization on the SM8250 feature design. It is expected that customers will not add any customizations within the Qualcomm delivered features, and utilize the provided customization hooks in case Qualcomm cannot enable such functionality within the feature due to some reason.
- Confirm feature combinations expected for SM8250. On SM8250, we plan to enable the following feature combinations:
  - Existing: MFNR + Bokeh, MFNR + QCFA
  - New: MFNR + in-sensor HDR, MFNR + Multi-frame HDR
- Identify feature-specific changes and work with Qualcomm to migrate to feature2.



Questions?

https://createpoint.qti.qualcomm.com