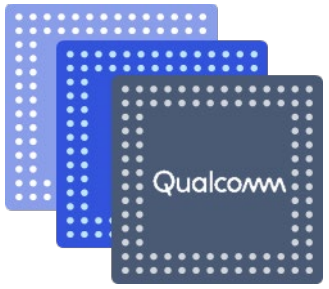
Qualcomm

Qualcomm Technologies, Inc.

SA8155/SA8195 Linux Android Automotive Camera Architecture Overview

80-PG469-95 Rev. B



Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

All Qualcomm products mentioned herein are products of Qualcomm Technologies, Inc. and/or its subsidiaries.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

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

Revision History

Revision	Date	Description
A	April 2019	Initial release
B	December 2019	There are many changes to this document, and it should be read in its entirety

Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

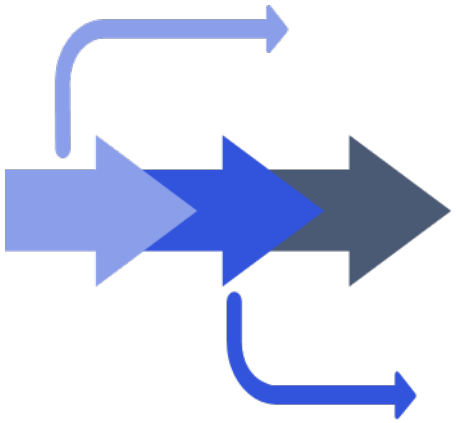
Contents

Document Scope

The purpose of this document is to provide an overview of the hardware capabilities of the camera system on SA8155/SA8195 SoC and software architecture. You can use this document to bring up the camera on the Qualcomm automotive development platform (ADP) with the AR0231 camera sensor

For bring-up procedures specific to a customized board with potentially different camera hardware, see *SA6155/SA8155 Automotive Camera AIS Customization Guide* (80-PG469-93)

Contents



Hardware Overview

Camera Hardware Blocks*

- Applications processor
  - Hosts Linux OS
  - Loads and executes camera driver software that controls all camera-related hardware blocks, sets up and controls clocks, as well as configures GPIO for camera sensor
- Camera Control Interface (CCI)
  - Used for dedicated I2C for camera subsystem
  - Commands are given CCI -> Deser via I2C, then Deser -> sensor via GMSL-2 or FPD-Link
- CSI PHY + CSID
  - Frame data entrance to SOC
  - Four 4-lane MIPI CSI configurable in 4 + 4 + 4 + 4 configuration
- Image Front End (IFE)
  - Hardware for dumping camera frame data to memory
  - Has four IFEs
  - Two IFEs are “Full IFE”s (3 RDI + 1 pipe for processing each)
  - Two are “Lite IFE”s (4 RDI each)
  - All paths through camera hardware will output to memory via an RDI (raw dump interface), regardless of processing needs
  - Processing happens after IFE dumps to memory
- Bayer Processing Engine (BPE)
  - Input is Bayer RAW frames from memory
  - Outputs YUV frames to memory
- Image Processing Engine (IPE)
  - Input is YUV frames from memory
  - Output is YUV frames to memory
  - Used for noise reduction

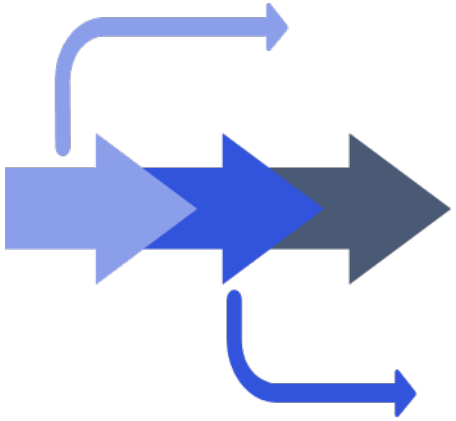
* Hardware capability does not guarantee software support.

Hardware Features in Camera*

Camera interfaces (CSID)	▪ 4 CSID ports ▪ MIPI Combo C/D-PHY 4/4/4/4 or 4/4/4/2/1 ▪ D-PHY – 2.5 Gbps/lane ▪ C-PHY – 5.71 Gbps/trio with three trios
Sensor driver support**	▪ LI-AR0231
Deserializer driver support**	▪ MAXIM 9296
IFEs	▪ 2x Lite IFE (760 MP/s each) ▪ 2x Full IFE (760 MP/s each)
Raw Dump Interfaces (RDI)	▪ Each Lite IFE can support up to 4 RDI streams simultaneously ▪ Each Full IFE can support up to 3 RDI streams simultaneously
ISP	▪ HDR via DSP/GPU ▪ Bayer processing via BPE ▪ Add'l post processing via BPE/IPE
Bayer sensors with HDR processing required	▪ 4 concurrent

* Hardware capability does not guarantee software support. Please check with local CE for clarification on software support.

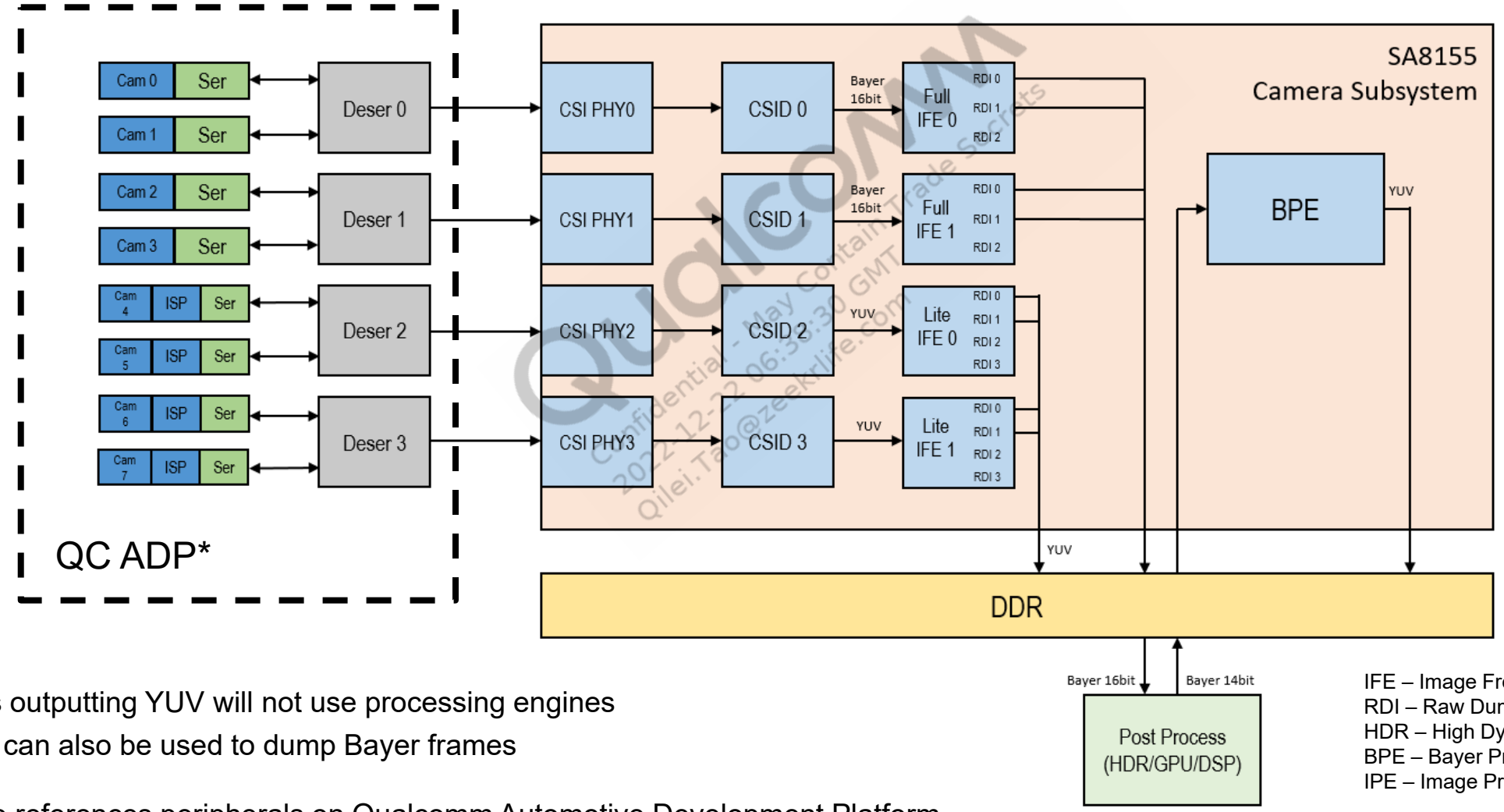
** Other sensors and deserializers are supported, but drivers are not provided.



Hardware Architecture

SA8155 Camera Subsystem

- Shown with 4 sensors outputting Bayer and 4 sensors outputting YUV



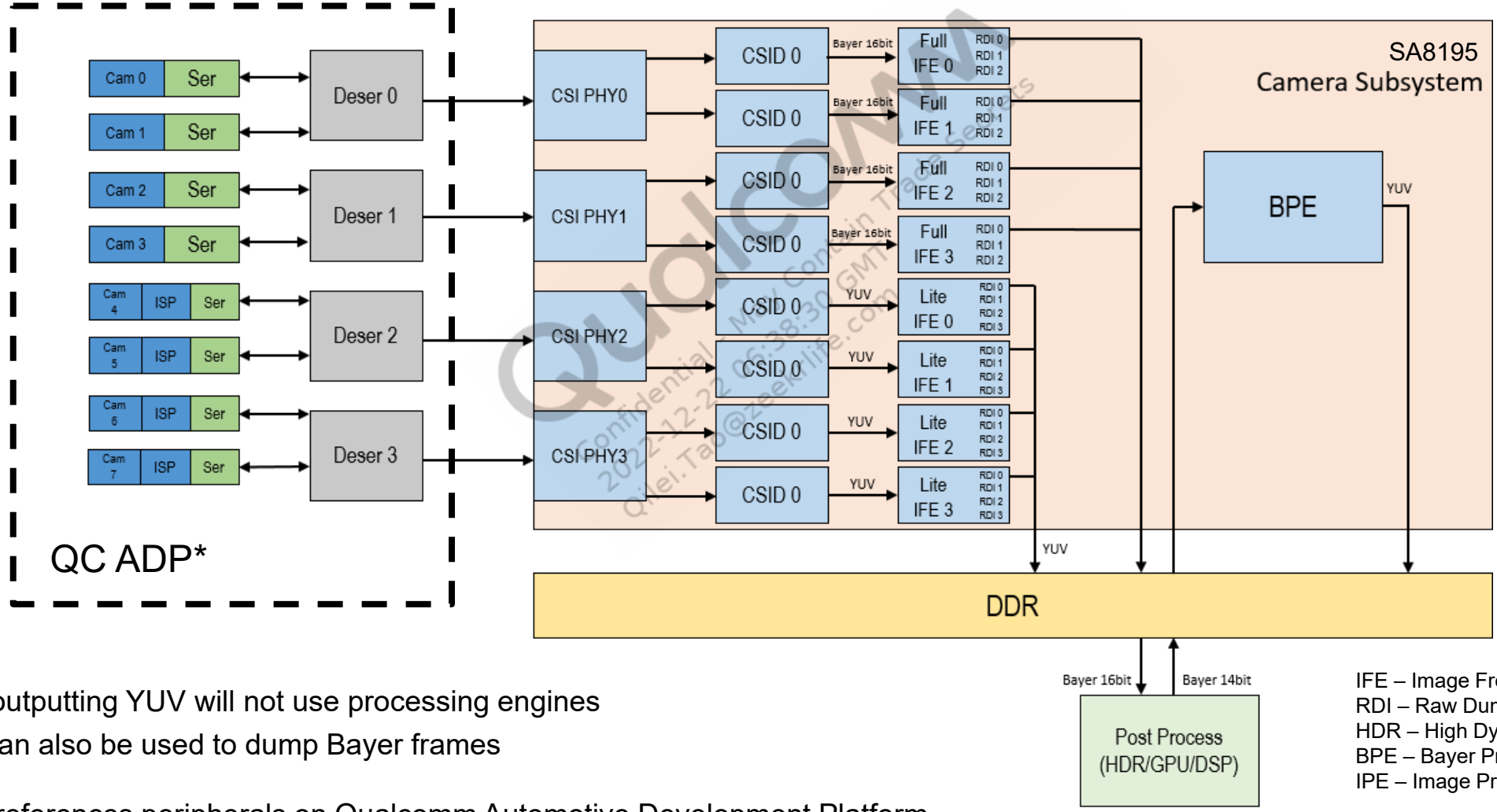
IFE – Image Front End
RDI – Raw Dump Interface
HDR – High Dynamic Range
BPE – Bayer Processing Engine
IPE – Image Processing Engine

- Sensors outputting YUV will not use processing engines
- Lite IFE can also be used to dump Bayer frames

* Example references peripherals on Qualcomm Automotive Development Platform.

SA8195 Camera Subsystem

- Shown with 4 sensors outputting Bayer and 4 sensors outputting YUV

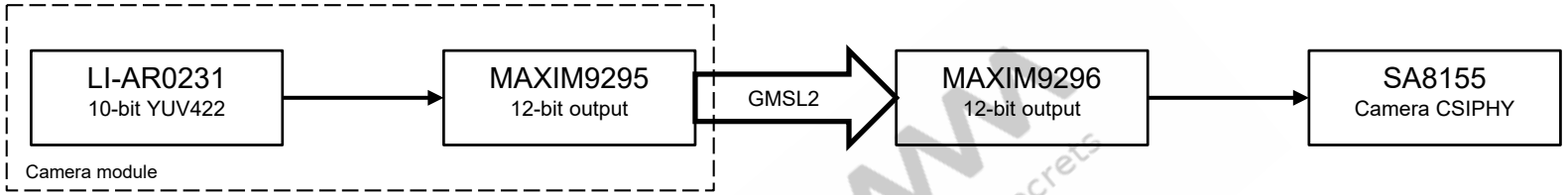


- Sensors outputting YUV will not use processing engines
- Lite IFE can also be used to dump Bayer frames

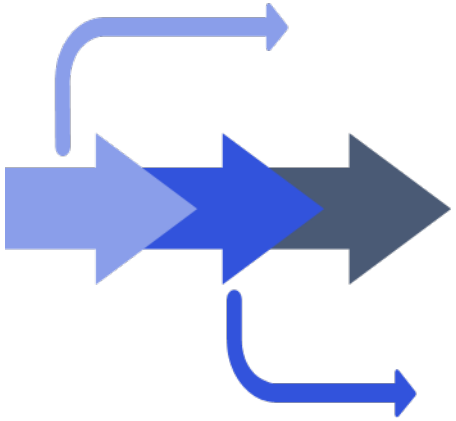
IFE – Image Front End
RDI – Raw Dump Interface
HDR – High Dynamic Range
BPE – Bayer Processing Engine
IPE – Image Processing Engine

* Example references peripherals on Qualcomm Automotive Development Platform.

Camera Components on SA8155/SA8195 ADP



- LI-AR0231 – Sensor, 10-bit YUV422 output
- MAXIM9295 – Serializer, 12-bit output
- LI-AR0231 and MAXIM9295 are part of the camera module
- GMSL2 – Fakra serial cable
- MAXIM9296 – Deserializer, 12-bit MIPI RAW output
- There are 4 MAXIM9296 deserializers on SA8155 ADP
- Up to 8 AR0231 sensors can be connected



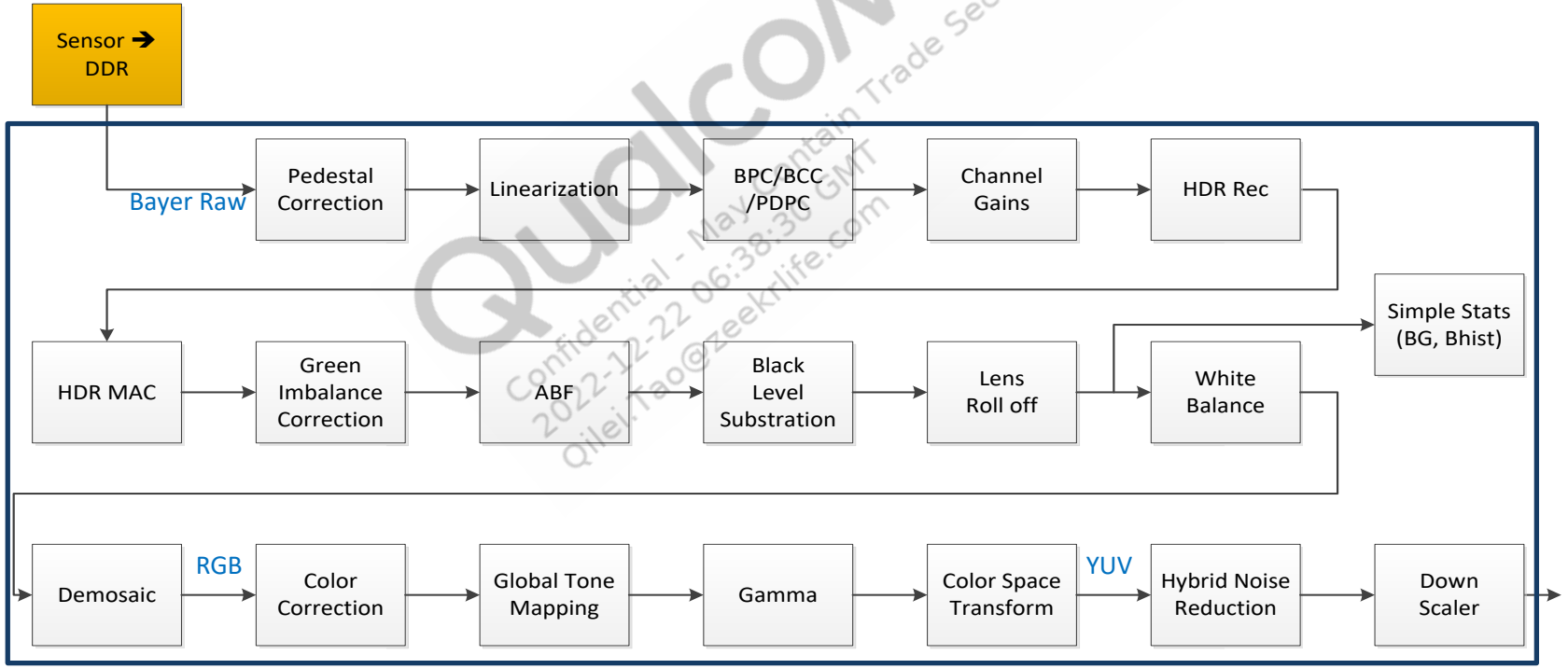
Internal ISP

ISP Overview

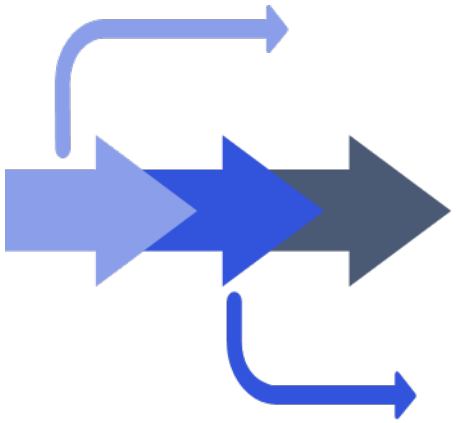
- All paths through camera hardware output to memory via a raw dump interface (RDI)
  - Regardless of processing needs, processing happens after the dump to memory
- HDR processing is done first by the GPU or DSP
- BPS is the processing segment that pulls frame data from memory, process, and then output back to memory
  - Responsible for conversion from Bayer color format to YUV

BPS Modules

- Each use-case has a defined path through the BPS called a pipeline
- Stats are used to tune ISP modules
- Pipelines defined in software via AIS engine -> CHI (camera hardware interface)



Note: Diagram shows hardware support. Contact local CE for confirmation on modules with software support.

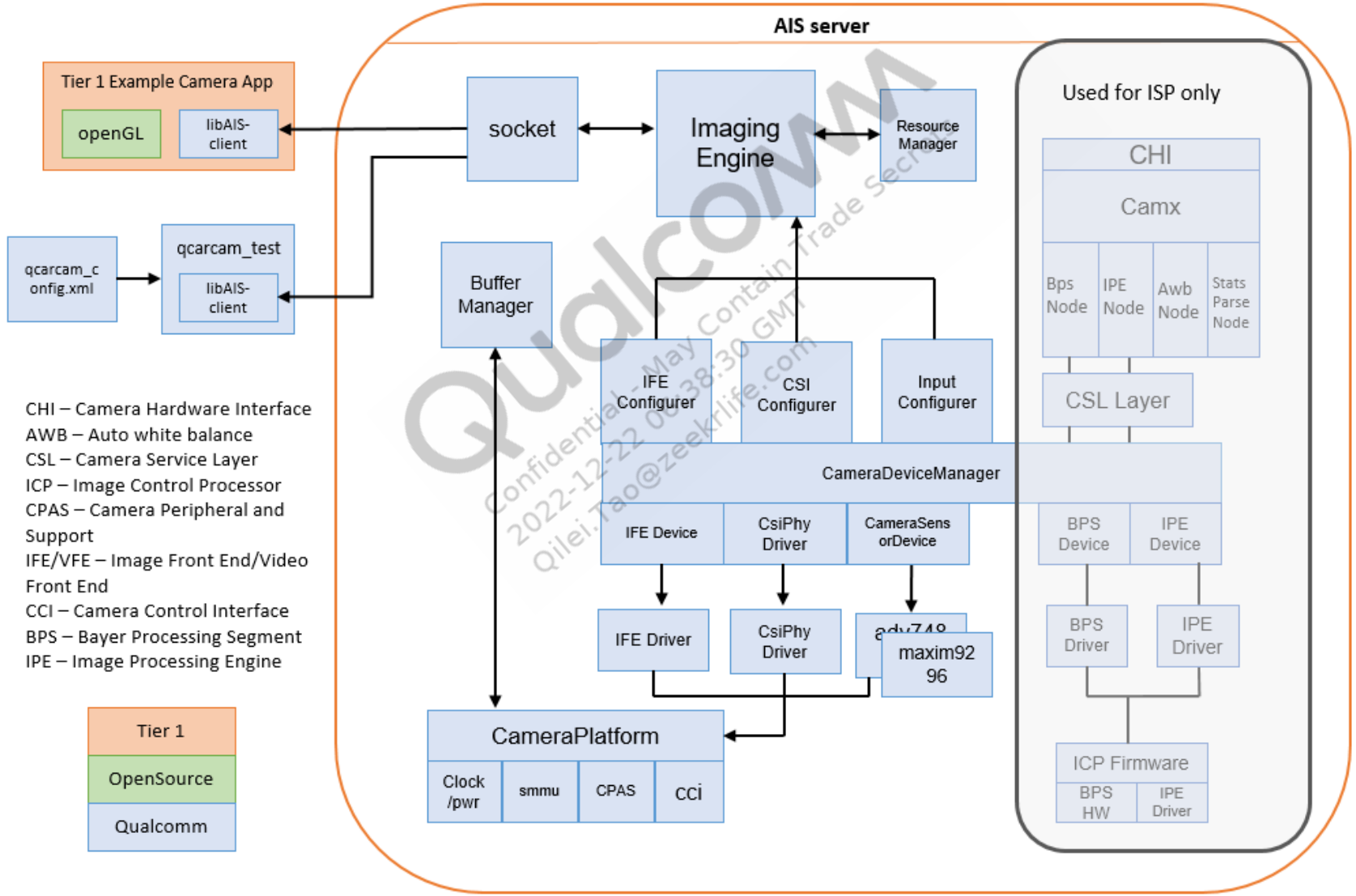


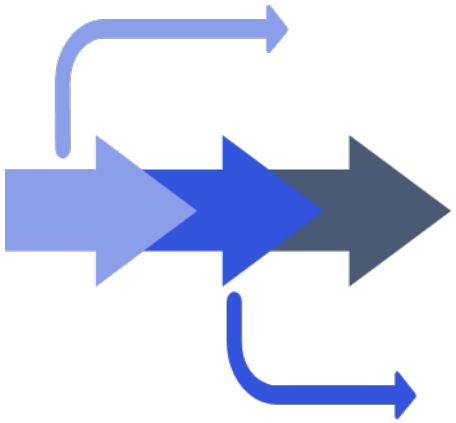
Software Architecture

Automotive Imaging System

- Automotive Imaging System (AIS) client/server architecture supports QCarCam, the QTI automotive proprietary API
- AIS is designed specifically for automotive use cases. This design includes OS portability, multi-user access, input permissions, and bridge chip abstraction for entity access

Automotive Imaging System (cont.)





QCarCam API

QCarCam API

QCarCam is the QTI automotive proprietary API. This section details QCarCam utility, benefits, and basic API calls

QCarCam call flow

QCarCam API – Test App

- The qcarcam_test native demo applications is executable over the terminal to quickly test the QCarCam API

Parameter	Description	Example
config	Specify qcarcam_config.xml file location	config=/bin/camera/qcarcam_test/qcarcam_config.xml
dumpFrame	Enable frame dump every X frames	dumpFame=50
startStop	Start/Stop every X frames	startStop=50
pauseResume	Pause/Resume every X frames	pauseResume=50
noDisplay	Run without displaying frames on the display	noDisplay
singlethread	Run qcarcam_test on a single thread	singlethread
printfps	Print average frames per second every X seconds	printfps=10

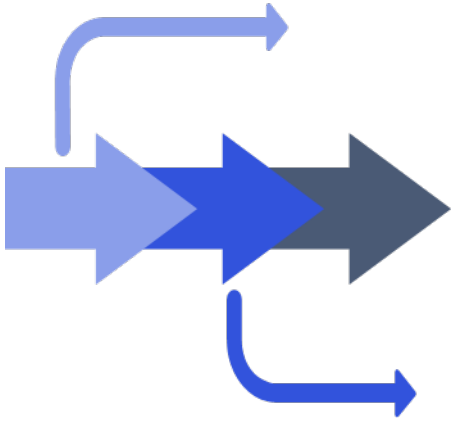
- To run on Android:

```
adb root
adb remount
adb shell
ais_server &
//one camera test for camera at channel 0
qcarcam_test -config=/system/bin/1cam.xml
//multi camera test can also be used for one camera
qcarcam_test -config=/system/bin/8cam.xml
```

- See release notes for latest qcarcam_test and config file details

Camera Bringup

- QCarCam API works out-of-the-box on Qualcomm ADPs with reference sensor. For use with customer platforms or alternative camera sensors, AIS customization is required
- See *SA6155/SA8155 Automotive Camera AIS Customization Guide* (80-PG469-93) for bringup procedures



Debugging

Debugging – Increasing Log Level

In vendor/qcom/proprietary/ais/Common/src/ais_log.c:

```
/**
* ais default configuration
*/
#if defined(__INTEGRITY)
- #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_CONSOLE, AIS_LOG_LVL_HIGH)
+ #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_CONSOLE, AIS_LOG_LVL_DBG2)
#elif defined(CAMERA_UNITTEST)
- #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_CONSOLE, AIS_LOG_LVL_HIGH)
+ #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_CONSOLE, AIS_LOG_LVL_DBG2)
#else
- #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_OS, AIS_LOG_LVL_HIGH)
+ #define AIS_LOG_DEFAULT_CONF AIS_LOG_CONF_MAKE(AIS_LOG_MODE_OS, AIS_LOG_LVL_DBG2)
#endif
```

Debugging – Validate i2c

Use ccidbgr to check i2c connection to deserializer and read/write

Location: /vendor/qcom/proprietary/ais/test/ccidbgr
Usage:

```
ais_server &
ccidbgr -dev=[slotId]  //slotId is the sensor slot id to be used. Default is 0.
```

Then follow the menu options. First set slave address [8bit format], address type [# of bytes], and data type [# of bytes]

```
cci_update
[slave addr] [addr_type] [data type]
```

Use read/write commands

```
cci_read
[addr]

cci_write
[addr] [data]
```

References

Documents	
Title	**Number**
Qualcomm Technologies, Inc.	
SA6155/SA8155 Automotive Camera AIS Customization Guide	80-PG469-93
QCarCam Automotive Camera API Reference for SA6155/SA6155P and SA8155/SA8155P	80-PK753-118

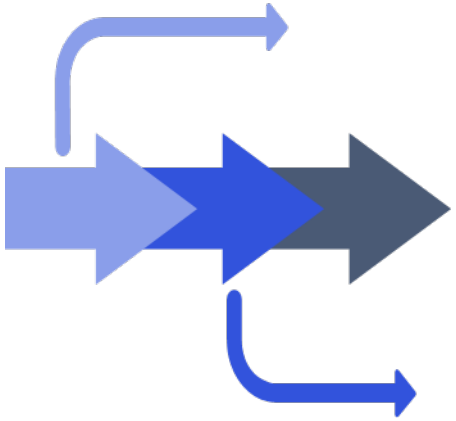
Acronyms	
Acronym or term	**Definition**
ABF	Adaptive Bayer Filter
ACE	Advanced Chroma Enhancement
ADP	Automotive Development Platform
AF	Auto Focus
ASF	Adaptive Spatial Filter
AWB	Auto White Balance
BLC	Black Level Correction
CAC	Chromatic Aberration Correction
CCI	Camera Control Interface
CCM	Color Correction Matrix
CCT	Color Correction Table
CPP	Camera Post Processing

References (cont.)

Acronyms	
Acronym or term	**Definition**
CS	Chroma Suppression
DMA	Direct Memory Access
DPC	Defective Pixel Correction
FD	Face Detection
FIR	Finite Impulse Response
GIC	Green Imbalance Correction
IIR	Infinite Impulse Response
LSC	Lens Shading Correction
LTM	Local Tone Mapping
MCTL	Media Control
DMA	Direct Memory Access
DPC	Defective Pixel Correction
FD	Face Detection
MCE	Memory Color Enhancement
MIPI	Mobile Industry Processor Interface
PDAF	Phase Detection Auto Focus
PVL	Preferred Vendor List
RNR	Radial Noise Reduction
RDI	Raw Dump Interface
SCE	Skin Color Enhancement
SNR	Skin Noise Reduction

References (cont.)

Acronyms	
Acronym or term	**Definition**
TNR	Temporal Denoise
VFE	Video Front-End of Camera Firmware
WNR	Wavelet Noise Reduction



Questions?

https://createpoint.qti.qualcomm.com