VISION SDK on BIOS (v02.07.00)

Data Sheet

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Table of Contents

1 Introduction

IMPORANT NOTE:

- This datasheet has performance and feature information about use-cases running on TDA2xx (EVM and Multi-sensor fusion platform a.k.aMonstercam platforms), TDA2Ex (EVM)andTDA3xx (EVM). Many use-cases can be run on multiple SOC/Platforms however the measurement of performance and other metrics is done only on one of these as indicated in the use-case details.

1.1 Framework features

- Compile and build for all CPUs
  - TDA2xx system (IPU1-0, IPU1-1, DSP1, DSP2, EVE1, EVE2, EVE3, EVE4, A15_0).
  - TDA3xx system (IPU1-0, IPU1-1, DSP1, DSP2, EVE1).
  - TDA2Ex system (IPU1-0, IPU1-1, DSP1, A15_0).
  - TDA3xx system (IPU1-0, IPU1-1, DSP1, DSP2, EVE1).
- Debug and release build profiles.
- Single place to setup memory map
  - default 1GB DDR memory map in TDA2xx
  - default 1GB DDR memory map in TDA2Ex
  - default 512MB DDR in TDA3xx
- Ability to create and control links on any CPU.
- Remote log feature with ability to print logs from all CPUs to UART controlled by IPU1-0.
- CPU load profiling for all cores.
- Memory usage log for all cores.
- Run time use-cases performance statistics like frame-rate, latency, frame-drop.
- Exception handler log for IPU, DSP.
- EDMA support on all cores, wherever applicable
  - A15, IPU uses system EDMA (via EDMA3LLD library)
  - DSP uses system EDMA or local EDMA (via EDMA3LLD library)
  - EVE uses local EDMA (via EDMA3LLD library and/or EVE SW library)
- Global timestamp to keep track of common time across all CPUs.
- Buffer allocation APIs for DDR, OCMC memory, L2 memory (DSP/EVE).
- Use-Case Auto Generation tool which generates C code for Vision SDK use-cases from config file.

1.2 Supported Links

No.	Links	TDA2xx-EVM	TDA2xx-MC	TDA2Ex	TDA3xx
Framework related links					
1	VIP Capture	√	√	√	√
2	Display	√	√	√	√
3	Display controller	√	√	√	√
4	IPC OUT/IN	√	√	√	√
5	Dup	√	√	√	√
6	Merge	√	√	√	√
7	Sync	√	√	√	√
8	Select	√	√	√	√
9	Null	√	√	√	√
10	Null Source	√	√	√	√
11	ISS Capture	X	X	X	√
12	ISS M2M ISP	X	X	X	√
13	ISS M2M Simcop	X	X	X	√
14	VPE	√	√	√	√
15	AVB receiver	√	√	X	X
16	MJPEG decoder	√	√	X	X
17	MJPEG encoder	√	√	X	X
18	H264 decoder	√	√	X	X
19	H264 encoder	√	√	X	X
20	Gate	√	√	√	√
Other Sample link's					
1	Graphics Source	√	√	√	√
2	Ultrasonic Capture	√	X	X	X
3	Network Ctrl	√	√	√	√
4	Split	√	√	√	√
Algorithm plugin's					
1	Alg plugin for Frame Copy	√	√	√	√
2	Alg plugin for Color to gray	√	√	√	√
3	Alg plugin for DMA Software	√	√	√	√

No.	Links	TDA2xx-EVM	TDA2xx-MC	TDA2Ex	TDA3xx
	Mosaic				
4	Alg plugin for Edge detection	√	√	X	√
5	Alg plugin for Dense Optical Flow	√	√	X	√
6	Alg plugin for Object detection	√	√	X	√
7	Alg plugin for Sparse optical flow	√	√	X	√
8	Alg plugin for SubFrame Copy	√	√	X	√
9	Alg plugin for ISS AEWB	X	X	X	√
10	Alg plugin for Synthesis	√	X	√	√
11	Alg plugin for Photometric Alignment	√	X	√	√
12	Alg plugin for Geometric Alignment	√	X	√	√
13	Alg plugin for Ultrasound	√	X	X	X
14	Alg plugin Soft ISP	X	√	X	X
15	Alg plugin Census	X	√	X	X
16	Alg plugin Disparity	X	√	X	X
17	Alg plugin Stereo Post Process	X	√	X	X
18	Alg plugin Remap Merge	X	√	X	X
19	Alg Plugin Lane Detect	√	√	√	√
20	Alg Plugin CRC	X	X	X	√

1.3 Features Not Supported and/or not Tested

- Exception handler log for EVE's
- Capture link
  - Multiple instances of capture link (NOTE: Single capture instance can control multiple HW VIP ports)
  - Enabling multiple VIP ports when subframe capture is enabled.
- VPE link
  - VPE link in DEI mode
  - Dual output mode – NOT supported by VPE HW
  - RGB input mode – NOT supported by VPE HW

- Algorithm link
    - Multi-channel mode NOT supported in Algorithm plugin’s
- Algorithm link: Subframe copy plugin
    - Multiple instances of this algplugin NOT supported
- TDA2xx-Monstercam
    - AVB not validated.
    - Linux not supported.

For detailed list refer to \docs\vision_sdk_feature_list.xlsx.

2 Single Channel Capture Use Case

2.1 Overview

This use case consists of continuous capture and display with optional frame copyusing DSP/EVE/A15 in between or edge detect algorithm using EVE.

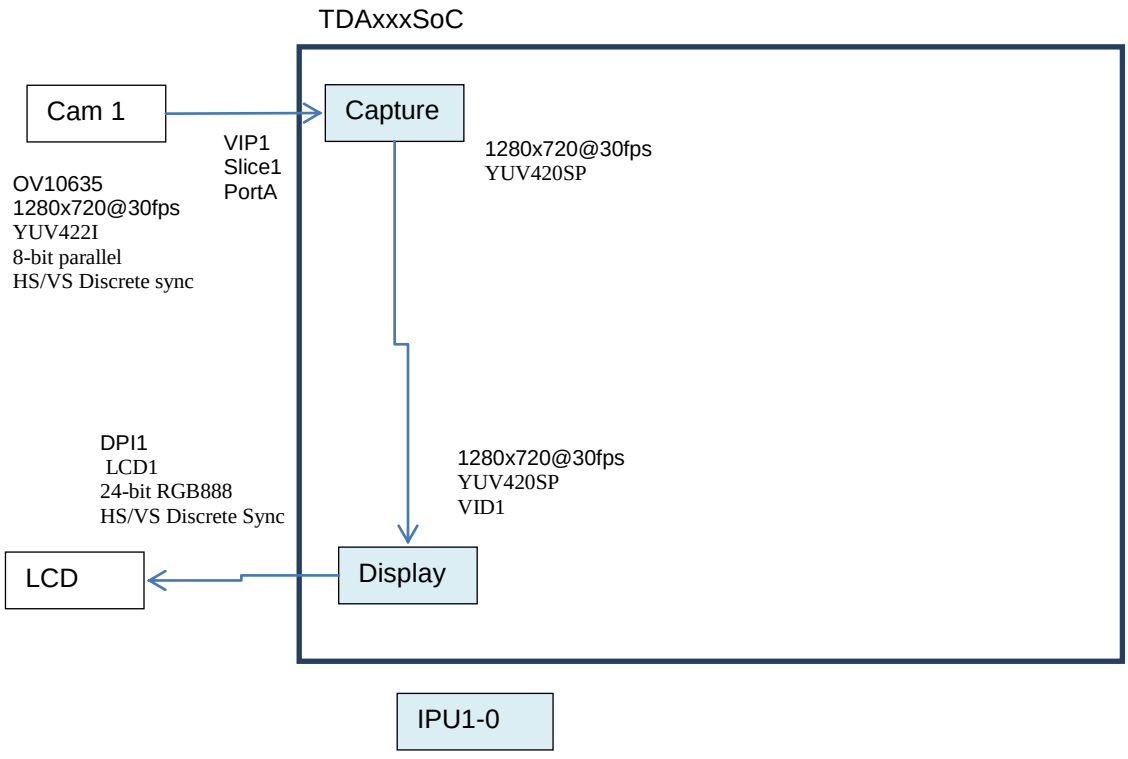
Frame copy/Edge Detect is used to show how an algorithm can be integrated in a capture to display data flow.

Capture can be done at 720p@30fps (OV Sensor)or 1080p@60fps (HDMI capture)via the VIP1 Slice1 PortA. Display can be on LCD via DPI1 output port or on HDMI display via HDMI output port. If required, a scalar is enabled as part of the capture path to match display resolution.The output interface runs at 60fps. In case incoming frames to display are at 30fps (when using OV Sensors), then the Display driver will repeat frames.

2.2 DataFlow

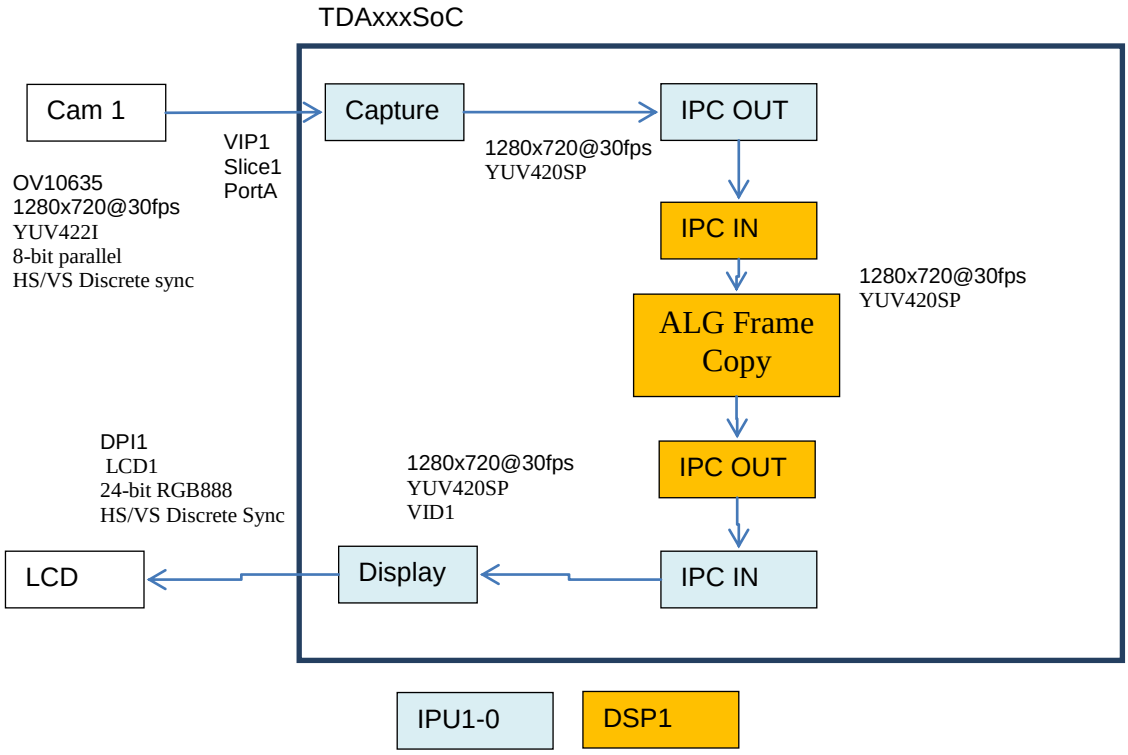
2.2.1 Configuration 1: Capture to Display on IPU 1-0

This configuration is useful for board checkout.Since only IPU1-0 CPU is required to run this use-case so it is easy to build and load and run. The data flow below shows a 720p30 capture with display on LCD. Alternately HDMI capture and HDMI display are also supported.



2.2.2 Configuration 2: Capture to Display via Frame Copy Algorithm DSP1

Frame copy algorithm on DSP uses DSP local EDMA for frame copy. EDMA3LLD library is used to access EDMA. The data flow below shows a 720p30 capture with display on LCD. Alternately HDMI capture and HDMI display are also supported.



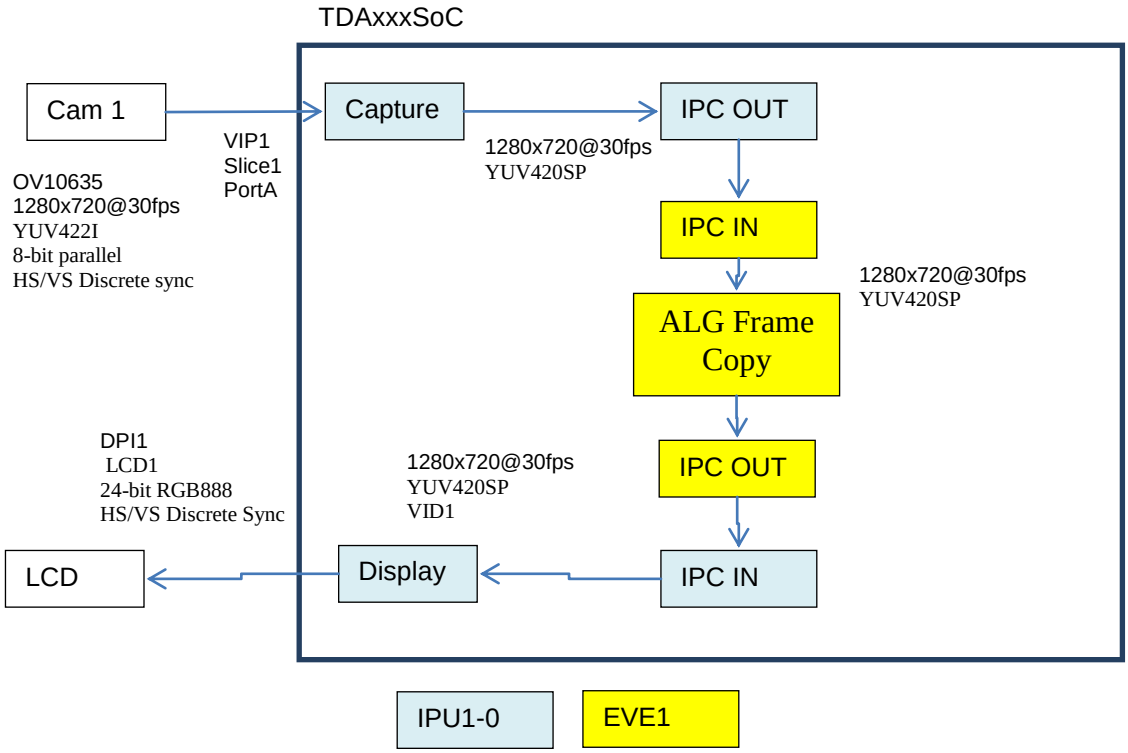
2.2.3 Configuration 3: Capture to Display via Frame Copy Algorithm EVE1

This is same as configuration 2 except EVE1 is used for frame copy instead of DSP1.

EVE1 uses its local DMA for frame copy. EVE SW library is used to do the EDMA.

EDMA3LLD is not used since normally algorithms on EVE will use the EVE SW library for EDMA.

The data flow below shows a 720p30 capture with display on LCD. Alternately HDMI capture and HDMI display are also supported.



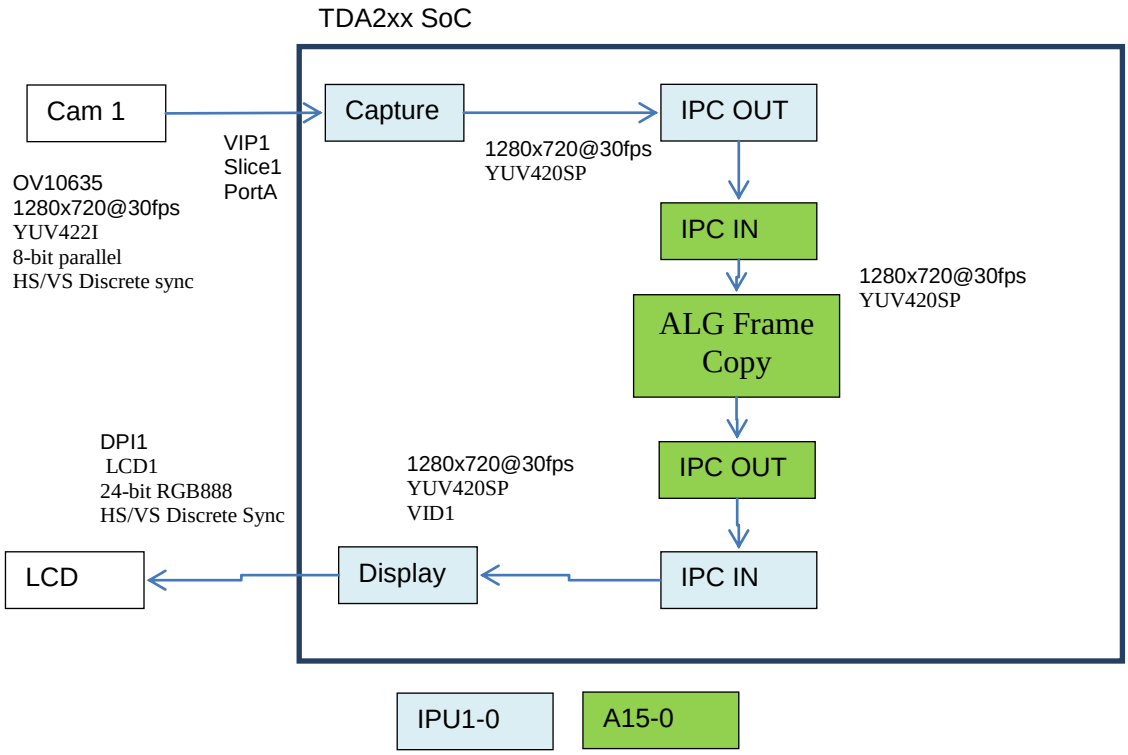
2.2.4 Configuration 4: Capture to Display via via Frame Copy Algorithm A15 (TDA2xx Only)

This is same as configuration 2 except A15 is used for frame copy instead of DSP1.

A15 uses CPU copy via cache for frame copy. Cache invalidates and write back are done as required in the process.

EDMA is not used since normally algorithms on A15 prefer to use CPU access via cache. System EDMA APIs via EDMA3LLD are however available for access via A15.

The data flow below shows a 720p30 capture with display on LCD. Alternately HDMI capture and HDMI display are also supported.



2.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for 720p30fps capture and HDMI display (1080p60) scenario.

2.4 CPU Loading and Task Info (TDA2xx)

2.4.1 Total CPU Load

NOTE: DSP and EVE use frame copy EDMA in polled mode, since normally in between DMA, algorithm computation is done.

NOTE: A15 uses frame copy in CPU mode (via cache) hence its CPU load is higher than DSP/EVE.

CPU	LOAD TYPE	CPU LOAD (CONFIGI1 IPU ONLY)	CPU LOAD (CONFIG2 DSP FC ALG)	CPU LOAD (CONFIG3 EVE FC ALG)	CPU LOAD (CONFIG4 A15 FC ALG)
IPU1-0	HWI	1.3%	1.6%	1.5%	1.5%
	SWI	0.3%	0.3%	0.3%	0.3%
	Total	**5.5%**	**8.0%**	**6.8%**	**7.6%**
DSP1	HWI	NA	0.2%	NA	
	SWI		0.0%		
	Total		**4.7%**		
EVE1	HWI	NA		0.6%	NA
	SWI			0.3%	
	Total			**5.9%**	
A15-0	HWI	NA			0.2%
	SWI				0.1%
	Total				**28.6%**

2.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (CONFIG1 IPU ONLY)	CPU LOAD (CONFIG 2 DSP FC ALG)	CPU LOAD (CONFIG 3 EVE FC ALG)	CPU LOAD (CONFIG 4 A15 FC ALG)
IPU 1-0	Stat Collector	Statistics collector	2.0%	2.1%	1.3%	1.6%
	Capture	Capture frames via VIP port	0.1%	0.1%	0.1%	0.1%
	Display	Display frames via DSS	0.3%	0.4%	0.8%	0.4%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	2.9%	4.8%	1.4%	4.9%
	IPC OUT	To send frame to another processor	NA	0.4%	0.4%	0.4%
	IPC IN	To receive frames from another processor		0.2%	0.2%	0.2%
DSP 1	*IPC + ALG Frame Copy	Frame copy + IPC	NA	4.2%	NA	
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load		0.5%		
EVE 1	*IPC + ALG Frame Copy	Framecopy + IPC	NA		5.0%	NA
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load			0.8%	
A15-0	IPC IN	To receive frames from another processor	NA			0.1%
	*IPC + ALG Frame Copy	Copy frames from input buffer to output buffer Using CPU copy via cache				26.8%
	IPC OUT	To send frame to another processor				0.1%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load				1.7%

*NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity.

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

2.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	CONFIG1 IPU ONLY	CONFIG2 DSP FC ALG	CONFIG3 EVE FC ALG	CONFIG4 A15 FC ALG
Capture	Output FPS	30fps	30fps	30fps	30fps
ALG Frame Copy (DSP / A15 / EVE)	Output FPS	NA	30fps	30fps	30fps
	Avg copy time per frame		1.4ms	1.3ms	9ms
Display	Input FPS	30fps	30fps	30fps	30fps
	VENC FPS	60fps	60fps	60fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The above benchmarks are computed for 720p30 fps captureand HDMI display (1080p60)

2.6 System Memory Usage (TDA2xx)

2.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage.

2.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED (CONFIG1 IPU ONLY)	MEMORY SIZE USED (CONFIG2 DSP FC ALG)	MEMORY SIZE USED (CONFIG3 EVE FC ALG)	MEMORY SIZE USED (CONFIG4 A15 FC ALG)
IPU1-0	Local Heap	256 KB	16KB	16KB	16KB	16KB
	HDVPSS Descriptor Mem	1MB	705KB	705KB	705KB	0 KB
DSP1	L2	223KB	NA	0 KB	NA	
	Local Heap	512 KB		23KB		
EVE1	L2	24 KB	NA		0 KB	NA
	Local Heap	256 KB			21 KB	
A15	Local Heap	4096KB	NA			57KB
Shared Memory	SR0 DDR	128KB	0KB	0KB	0 KB	0KB
	SR1 Frame Buffer	256MB	256 MB	0 MB	0 KB	0 KB
	SR2 OCMC	512 KB	0 KB	0 KB	0 KB	0 KB
	Remote Log Buffer	160KB	158 KB	158 KB	158 KB	0KB

NOTE:

- SR1 Frame Buffer now static memory allocation .map file show all memory used but in actual we don't know how much memory used. This is applicable all usecase.

2.7 Other Benchmarks (TDA2xx)

2.7.1 Processing Latency

		LATENCY (CONFIG1 IPU ONLY)	LATENCY (CONFIG2 DSP FC ALG)	LATENCY (CONFIG3 EVE FC ALG)	LATENCY (CONFIG4 A15 FC ALG)
Capture to Display Latency	Avg	0.56 ms	2.2ms	2.5 ms	9.8ms
	Min	0.06 ms	1.7ms	1.9 ms	9.3ms
	Max	61 ms	57.9ms	54ms	54.7ms

NOTE:

- This latency is as measured inside the system by software.

- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

2.7.2 Boot Time

PARAMETER	VALUE
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059s
main() to Use-case create	0.340s
Use-case create start to Live preview on display (max value from all 4 configurations)	0.065 s
Total Boot time	**2.464s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

3 Subframe copy Use case

3.1 Overview

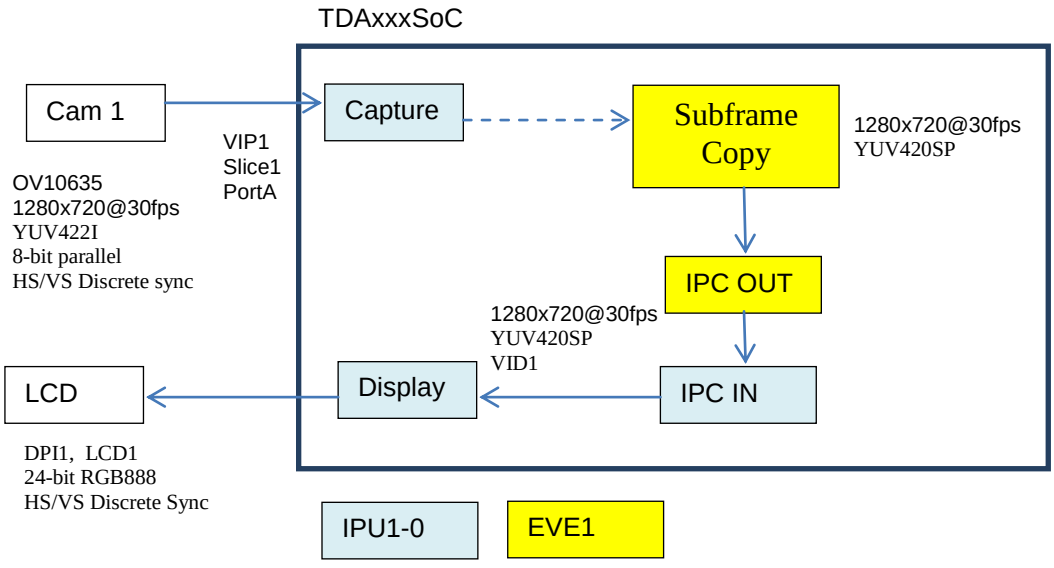
This configuration is used to demonstrate a low-latency path which utilizes the subframe capture feature.

3.2 DataFlow

3.2.1 Subframe copy example

The Capture can be either OV10635 Sensor @720p30 or HDMI @1080p60 via the VIP1 Slice1 PortA. Scalar is required to be enabled for this feature.The Capture link enables subframe-capture mode and configures the VIP but does not process any frame data. Hence no IPC link is required between the capture link on IPU1_0 and Subframe copy link on EVE1. The Subframe copy link registers interrupts with VIP at subframe and frame boundaries. Whenever a sub-frame is received an EDMA transfer for the data is initiated onto display buffer. Subframe copy algorithm on EVE uses local EDMA for subframe copy. EDMA3LLD library is used to access EDMA. The display link processes data at frame boundaries. Display can be on LCD via DPI1 output port or on HDMI display via HDMI output port.The output interface runs at 60fps. In case incoming frames to display are at 30fps (when using OV Sensors), then the Display driver will repeat frames.

The data flow below shows OV Sensor capture and LCD display. Alternately HDMI capture and HDMI display are also supported.



3.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for OV10635capture and HDMI display (1080p60) scenario.

3.4 CPU Loading and Task Info (TDA2xx)

3.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	**HWI**	**1.3%**
	SWI	**0.5%**
	Total	**6.9%**
EVE1	**HWI**	**2.3 %**
	SWI	**0.2%**
	Total	**7.5%**

3.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
IPU1-0	Stat Collector	Statistics collector	1.3%
	Capture	Capture frames from sensor via VIP port	0.0 %
	Display	Display via DSS	0.4%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	2.9%
	IPC IN	To receive frames from another processor	0.2%
*EVE1	IPC +ALG Subframe copy	Algsubframe copy + IPC	5.3 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	2.3 %

*NOTE: On EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load calculations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

3.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	
ALG-SubframeCopy (EVE1)	Input fps	30fps
	Output fps	30fps
Display	Input fps	30fps
	VENC fps	60fps

NOTE: The ALG subframe copy link processes ISRs from VIP for subframe and frame complete events. When the alg plugin link is in running state no other command except data command is processed, to ensure the low latency path functions effectively. Hence the stats for this link are not queries/printed on UART. The figures mentioned above have been interpreted from other logs and performance of IPC and Display links.
NOTE: FPS numbers are rounded off to nearest integer

3.6 System Memory Usage (TDA2xx)

3.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

3.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	15KB
	HDVPSS Descriptor Mem	1MB	705KB
EVE1	L2	24 KB	0 KB
	Local Heap	256 KB	13 KB
Shared Memory	SR0	128 KB	0 MB
	Frame Buffer (SR1)	255MB	12 MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160 KB	158 KB

Note :Subframes are allocated from OCMC RAM1 - 512KB

3.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	
EMIF Read Only	Avg	348 MB/s
	Peak	477 MB/s
EMIF Write Only	Avg	46 MB/s
	Peak	86 MB/s
EMIF Read + Write	Avg	394 MB/s
	Peak	477 MB/s

3.7 Other Benchmarks (TDA2xx)

3.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 1)	Avg	0.8ms
	Min	0.4 ms
	Max	57 ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

3.7.2 Boot Time

PARAMETER	DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s
main() to Use-case create	0.340s
Use-case create start to Live preview on display	0.050 s
Total Boot time	**2.449s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

4 Dense Optical Flow Usecase

4.1 Overview

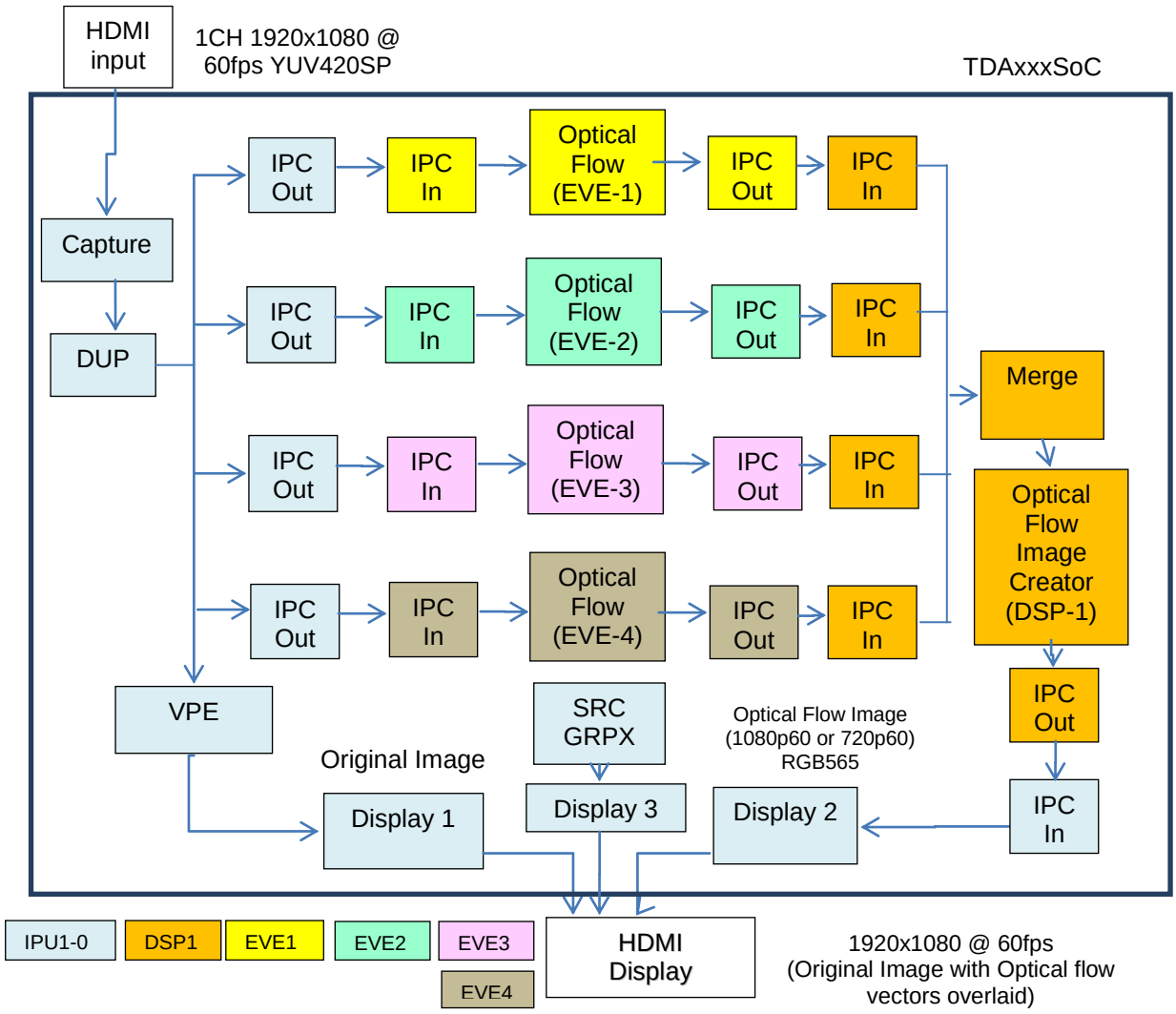
This configuration is used to demonstrate the capability of EVE. The display type is restricted to HDMI alone.

4.2 DataFlow

4.2.1 Dense Optical Flow example

The Capture can be either OV10635 Sensor @720p30 or HDMI @1080p60. The captured buffer is duplicated by DUP link running on IPU1-0 and sent to all 4 EVEs. A Dense optical flow algorithm runs on the EVEs. An algorithm setting is available by which one can choose to run a 1 pyramid (1080p60) or 2 pyramid (720p60) mode. All 4 EVEs are used in this use-case for frame processing. Each EVE processes 1 out of 4 frames. The output from the 4 EVEs is merged on DSP1 and post an Optical Flow Image creator sent to HDMI display. The original image after scaling using VPE is also displayed alongwith some overlaid graphics.The data flow below shows an HDMI capture. Alternately OV Sensor capture is also supported. This configuration supports ONLY HDMI display.

NOTE: VPE, EVE2, EVE3, EVE4 only valid in TDA2xx



4.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for HDMI capture (1080p60) and HDMI display (1080p60) scenario.

4.4 CPU Loading and Task Info (TDA2xx)

4.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD (CONFIG - ALG 1 PYRAMID)	CPU LOAD (CONFIG - ALG 2 PYRAMID)
IPU1_0	**HWI**	**4.2 %**	**4.0**
	SWI	**0.8 %**	**0.8**
	Total	**17.5 %**	**17.3** %
DSP1	**HWI**	**0.3**	**0.4**
	SWI	**0.1**	**0.1**
	Total	**64.7 %**	**27.6**%
EVE1	**HWI**	**1.1**	**1.0**
	SWI	**0.2**	**0.1**
	Total	**78.4 %**	**53.3%**
EVE2	**HWI**	**1.2**	**1.2**
	SWI	**0.6**	**0.1**
	Total	**78.5 %**	**56.6%**
EVE3	**HWI**	**1.2**	**1.0**
	SWI	**0.2**	**0.1**
	Total	**78.3 %**	**53.3 %**
EVE4	**HWI**	**1.2**	**1.0**
	SWI	**0.2**	**0.1**
	Total	**78.2 %**	**54.4 %**

4.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (CONFIG ALG 1 PYRAMID)	CPU LOAD (CONFIG ALG 2 PYRAMID)
IPU1-0	Stat Collector	Statistics collector	1.9 %	2.0 %
	Capture	Capture frames from sensor via VIP port	0.4 %	0.3 %
	Display1	Display scaled frames via DSS	0.9 %	0.8 %
	Display2	Display DOF frame via DSS	0.7 %	0.7 %

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (CONFIG ALG 1 PYRAMID)	CPU LOAD (CONFIG ALG 2 PYRAMID)
	DUP	Duplicate frame to send to display without scaling as well as to VPE to scale	0.7 %	0.6 %
	VPE	Frame scaling	2.0 %	2.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	7.3 %	7.0 %
	IPC OUT 0	To send frame to another processor	0.8 %	0.8 %
	IPC OUT 1	To send frame to another processor	0.6 %	0.7 %
	IPC OUT 2	To send frame to another processor	0.6 %	0.7 %
	IPC OUT 3	To send frame to another processor	0.6 %	0.6 %
	IPC IN	To receive frames from another processor	0.5 %	0.5 %
DSP1	IPC + ALGVector to Image	To receive frames from another processor	64.1 %	27.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.7 %	0.3 %
EVE1	IPC + Optical Flow	IPC Load + Dense Optical Flow Algorithm	76.6 %	54.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.5 %	1.3 %
EVE2	IPC + Optical Flow	IPC Load + Dense Optical Flow Algorithm	76.4 %	55.1 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.6 %	1.3 %
EVE3	IPC + Optical Flow	IPC Load + Dense Optical Flow Algorithm	76.6 %	54.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.6 %	1.3 %
EVE4	IPC + Optical Flow	IPC Load + Dense Optical Flow Algorithm	76.2%	55 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.5 %	1.2 %

NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

4.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	(CONFIG ALG 1 PYRAMID)	(CONFIG ALG 2 PYRAMID)
Capture	Output FPS	60fps	60fps
ALG DOF (EVE)	Output FPS	15 fps per EVE	15 fps per EVE
	Avg time per frame	54ms	36ms
ALG Vector to Image (DSP1)	Output FPS	60fps	60fps
	Avg time per frame	11ms	5ms
Display (DOF image)	Input FPS	60fps	60fps
	VENC FPS	60fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The figures are for HDMI capture scenario.

4.6 System Memory Usage (TDA2xx)

4.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

4.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED (CONFIG ALG PYRAMID 1)	MEMORY SIZE USED (CONFIG ALG PYRAMID 2)
IPU1-0	Local Heap	256 KB	15KB	15KB
	HDVPSS Descriptor Mem	1MB	705KB	705KB
DSP1	L2	223KB	109 KB	45 KB
	Local Heap	512 KB	13 KB	13 KB
EVE1	L2	24 KB	3 KB	3 KB
	Local Heap	256 KB	10 KB	10 KB
EVE2	L2	24 KB	3 KB	3 KB
	Local Heap	256 KB	10 KB	10 KB
EVE3	L2	24 KB	3 KB	3 KB
	Local Heap	256 KB	10 KB	10 KB
EVE4	L2	24 KB	3 KB	3 KB
	Local Heap	256 KB	10 KB	10 KB
Shared Memory	SR0	128KB	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB	256MB
	SR2 OCMC	512 KB	0 KB	0 KB
	Remote Log Buffer	160KB	158KB	158KB

4.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	ALG PYRAMID 1	ALG PYRAMID 2
EMIF Read Only	Avg	1167 MB/s	937 MB/s
	Peak	1448 MB/s	1428 MB/s
EMIF Write Only	Avg	653 MB/s	325 MB/s
	Peak	779 MB/s	552 MB/s
EMIF Read + Write	Avg	1819 MB/s	1263 MB/s
	Peak	2130 MB/s	1841 MB/s

4.7 Other Benchmarks (TDA2xx)

4.7.1 Processing Latency

		LATENCY (CONFIG ALG PYRAMID 1)	LATENCY (CONFIG ALG PYRAMID 2)
Capture to Display Latency (Display VID 2)	Avg	67 ms	43 ms
	Min	66 ms	42 ms
	Max	75 ms	60 ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

4.7.2 Boot Time

PARAMETER	DURATION (CONFIG 6 ALG PYRAMID 1)	DURATION (CONFIG 6 ALG PYRAMID 2)
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s	2.059 s
main() to Use-case create	0.340 s	0.340 s
Use-case create start to Live preview on display	0.456 s	0.255 s
Total Boot time	**2.855s**	**2.654s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

5 Sparse Optical Flow Use case

5.1 Overview

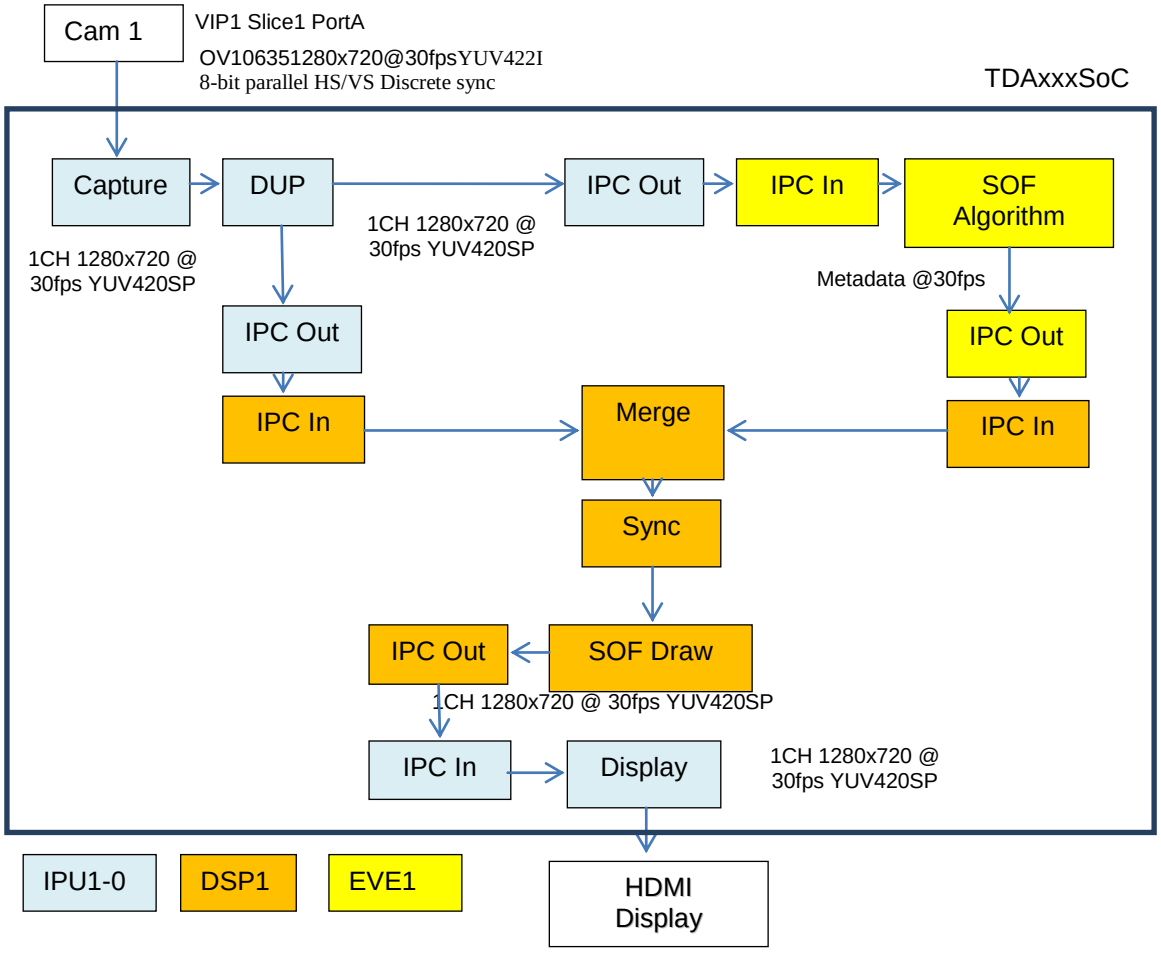
This configuration is used to demonstrate the sparse optical algorithm running on EVE and DSP.

5.2 DataFlow

5.2.1 Sparse Optical Flow example

The Capture can be either OV10635 Sensor @720p30 or HDMI @1080p60. The captured buffer is duplicated by DUP link running on IPU1-0. One output of DUP goes to a sparse optical flow algorithm running on EVE1. This generates metadata output which is subsequently merged along-with the original captured video frames and synced on DSP.The output of sync link is a composite buffer consisting of two channels - Channel 0 consists of original video frame buffer while channel 1 consists of metadata. The SOF draw link running on DSP takes this as input and draws feature point tracking on the video to depict the optical flow. This frame is then fed to the display

The data flow below shows OV Sensor capture and HDMI display. Alternately HDMI capture and LCD display are also supported.



5.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for 720p30 capture and HDMI display (1080p60) scenario.

5.4 CPU Loading and Task Info (TDA2xx)

5.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	**HWI**	**1.8**
	SWI	**1.1**
	Total	**10.0**
DSP1	**HWI**	**0.4**
	SWI	**0.1**
	Total	**4.2 %**
EVE1	**HWI**	**0.8**
	SWI	**0.2**
	Total	**66.9 %**

5.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
IPU1-0	Stat Collector	Statistics collector	2.0 %
	Capture	Capture frames from sensor via VIP port	0.2 %
	Display	Display via DSS	0.4 %
	DUP	Duplicate frame to send to display without scaling as well as to VPE to scale	0.2 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	4.4 %
	IPC OUT 0	To send frame to another processor	0.5 %
	IPC OUT 1	To send frame to another processor	0.3 %
	IPC IN	To receive frames from another processor	0.2 %
*DSP1	ALG SOF Draw + Sync + Merge + IPC	Major processing on DSP	3.5%

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.7%
*EVE1	IPC + Optical Flow	Sparse Optical Flow Algorithm + IPC	65.7%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.2%

*NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

5.5 System Performance

COMPONENT	PARAMETER	
Capture	Output FPS	30fps
ALG SOF (EVE1)	Output FPS	30fps
	Avg time per frame	15ms
ALG SOF Draw (DSP1)	Output FPS	30fps
	Avg time per frame	8ms
Display (DOF image)	Input FPS	30fps
	VENC FPS	60fps

NOTE: FPS numbers are rounded off to nearest integer

5.6 System Memory Usage (TDA2xx)

5.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

5.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	15KB
	HDVPSS Descriptor Mem	1MB	705 MB
DSP1	L2	223KB	0 KB
	Local Heap	512 KB	12 KB
EVE1	L2	24 KB	13 KB
	Local Heap	256KB	26 KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158 KB

5.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	DOF
EMIF Read Only	Avg	622 MB/s
	Peak	1409 MB/s
EMIF Write Only	Avg	147 MB/s
	Peak	703 MB/s
EMIF Read + Write	Avg	770 MB/s
	Peak	2080 MB/s

5.7 Other Benchmarks (TDA2xx)

5.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 1)	Avg	30ms
	Min	12ms
	Max	102ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.

- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

5.7.2 Boot Time

PARAMETER	DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s
main() to Use-case create	0.340 s
Use-case create start to Live preview on display	0.382 s
Total Boot time	**2.781 s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

6 Object(Pedestrian and Traffic Sign) Detect Use case

6.1 Overview

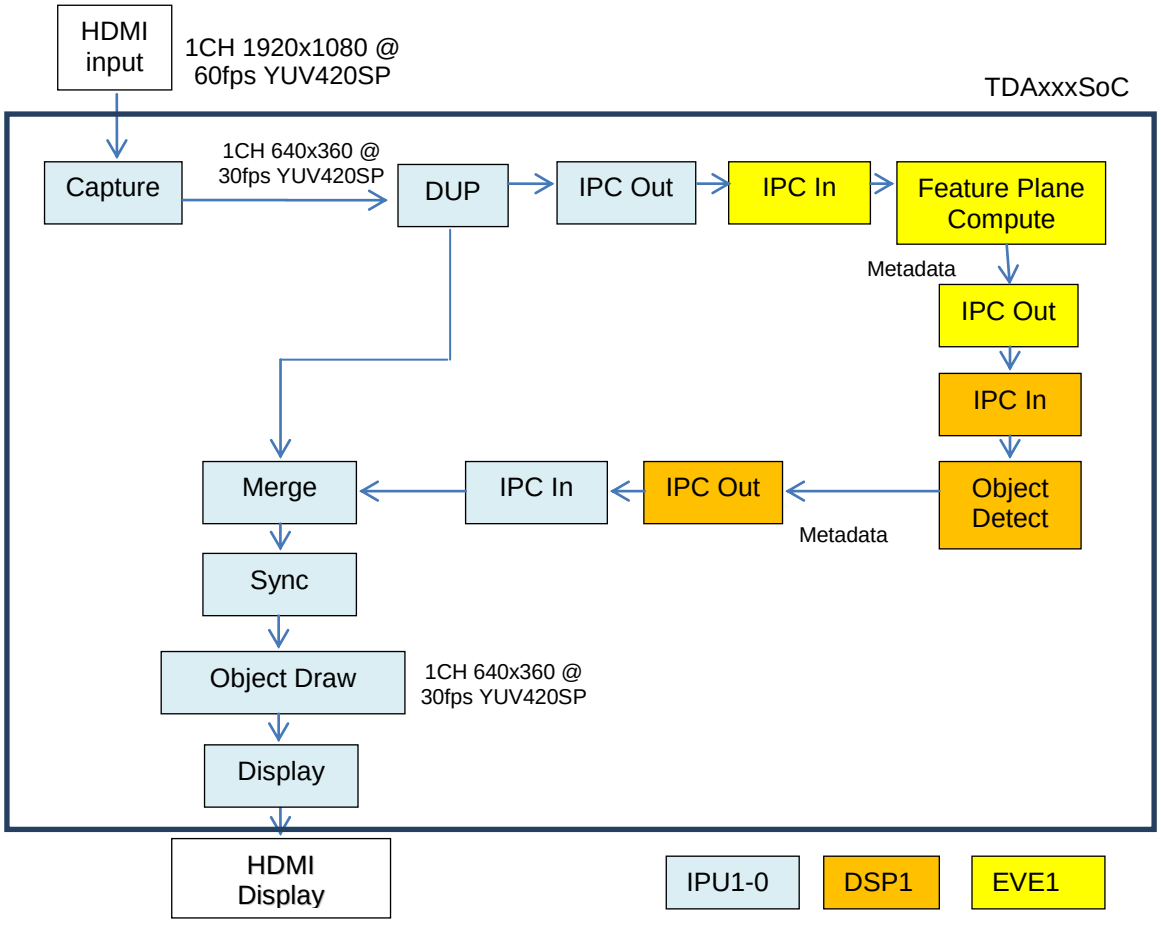
This configuration is used to demonstrate the capability of EVE and DSP in running aobject detection (Pedestrian and traffic sign) algorithm.

6.2 DataFlow

6.2.1 Object Detect example

The Capture is HDMI@1080p60 (pre-recorded data). The capture output is configured to 640x360 @30fps. This captured buffer is duplicated by DUP link running on IPU1-0 and sent to Merge link. A feature plane compute runs on EVE1 which generates metadata output buffer. The Object Detect link running on DSP utilizes this metadata information to generate co-ordinates of objectsi.e. pedestrians or traffic signs are present in the frame. These co-ordinates are merged and synced with captured frames on IPU1_0. The output of sync link is a composite buffer consisting of two channels - Channel 0 consists of video frame buffer while channel 1 consists of meta data. This buffer is processed by Object draw algorithm plugin link. If the identified object is a pedestrian then it draws a rectangle around the detected pedestrian. If the identified object is a traffic sign, it finds corresponding bitmap and draws it adjacent to the sign location. This frame is then fed to the display, where it is up-scaled based on display type.

The data flow below shows HDMI capture and HDMI display. Alternately LCD display arealso supported.



6.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for HDMI capture (1080p30) and HDMI display (1080p60) scenario.

6.4 CPU Loading and Task Info (TDA2xx)

6.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	**HWI**	**2.1**
	SWI	**0.7**
	Total	**12.4 %**
DSP1	**HWI**	**0.3**
	SWI	**0.1**
	Total	**36.0 %**
EVE1	**HWI**	**0.8**
	SWI	**0.4**
	Total	**65 %**

6.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
IPU1-0	Stat Collector	Statistics collector	2.1 %
	Capture	Capture frames from sensor via VIP port	0.2%
	DUP	Duplicate frame to send to merge link	0.1 %
	Sync	Sync frames based on timestamp from multiple channels	0.5%
	Merge	Merge frames from SW Mosaic and original capture output	0.2%
	Display1	Display scaled frames via DSS	0.7%
	ALG – Object Draw	Object detection algorithm	5.4%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	4.3%
	IPC OUT	To send frame to another processor	0.3%
	IPC IN	To receive frames from another processor	0.2%
DSP1	IPC + ALG Object Detect	IPC+ Object detection algorithm	35.4%

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.2 %
EVE1	ALG – Feature Plane Compute + IPC	Feature plane compute algorithm + IPC	64.1%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.2 %

NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

6.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	
Capture	Output fps	30fps
ALG-Feature Plane compute (EVE1)	Avg time per frame	31ms
	Output fps	30fps
ALG- Object Detect (DSP1)	Avg time per frame	27ms
	Output fps	30fps
ALG Object Draw (IPU1-0)	Avg time per frame	1ms
	Output fps	30fps
Display	Input fps	30fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The figures are for HDMI capture scenario.

6.6 System Memory Usage (TDA2xx)

6.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

6.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	17 KB
	HDVPSS Descriptor Mem	1MB	705KB
DSP1	L2	223 KB	224 KB
	Local Heap	512 KB	13 KB
EVE1	L2	24 KB	19 KB
	Local Heap	256KB	64KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256 MB	256 MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158KB

6.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	DOF
EMIF Read Only	Avg	772 MB/s
	Peak	1585 MB/s
EMIF Write Only	Avg	200 MB/s
	Peak	483 MB/s
EMIF Read + Write	Avg	973 MB/s
	Peak	2011 MB/s

6.7 Other Benchmarks (TDA2xx)

6.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 2)	Avg	62ms
	Min	57ms
	Max	98ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.

- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

6.7.2 Boot Time

PARAMETER	DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s
main() to Use-case create	0.340 s
Use-case create start to Live preview on display	0.282s
Total Boot time	**2.681sec**

GUI and Sensor initialization time not accounted for.

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

7 Lane Detect Use case

7.1 Overview

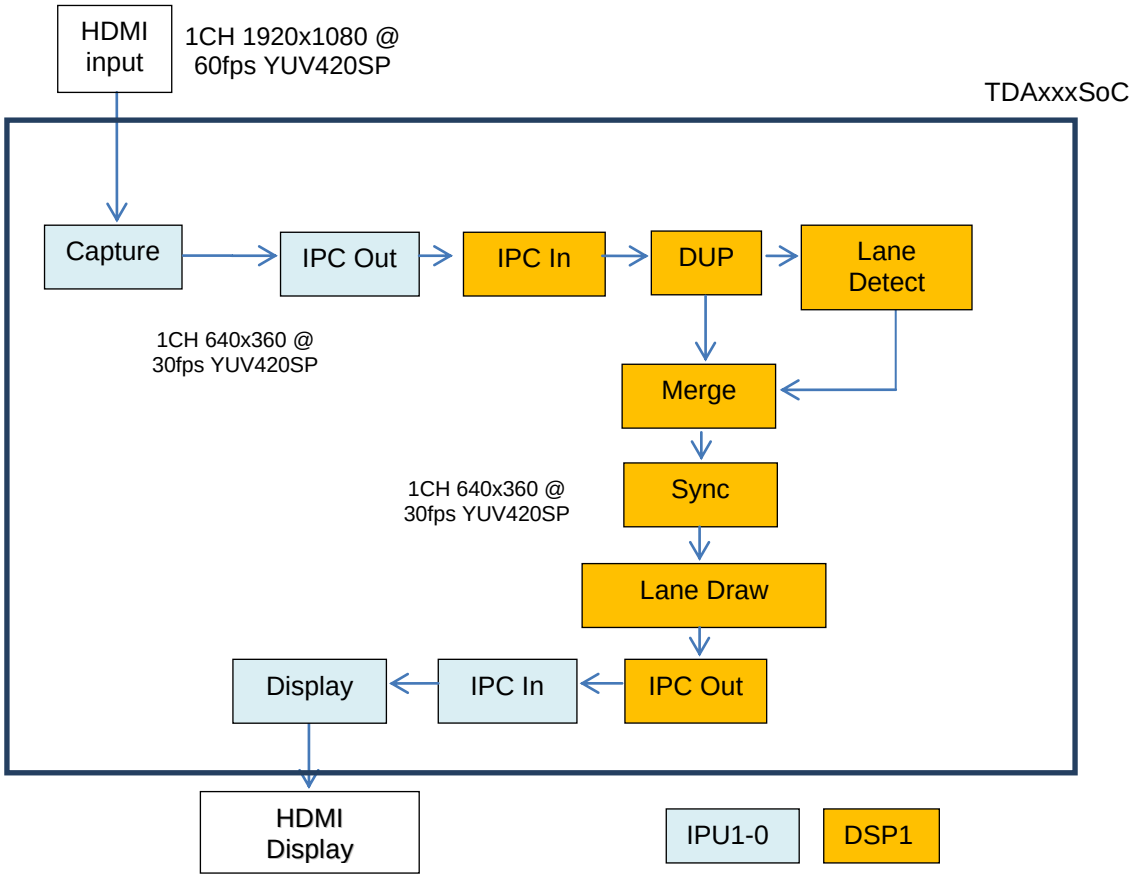
This configuration is used to demonstrate the capability of DSP in running a lane detection algorithm.

7.2 DataFlow

7.2.1 Lane Detect example

The Capture is HDMI @1080p60 (pre-recorded data). The capture output is configured to 640x360@30fps. This captured buffer is duplicated by DUP link running on DSP1 and one of the output's is sent to Merge link. A lane detect algorithm runs on DSP1 which generate co-ordinates of where lanes are present in the frame. These co-ordinates are merged and synced with captured frames on DSP1. The output of sync link is a composite buffer consisting of two channels - Channel 0 consists of video frame buffer while channel 1 consists of meta data. This buffer is processed by Lanedetect draw algorithm plugin, which draws a linewhere lane is present. This frame is then fed to the display, where it is up-scaled based on display type.

The data flow below shows HDMI capture and HDMI display. Alternately LCD display are also supported.



7.3 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for HDMI capture (1080p60) and HDMI display (1080p60) scenario.

7.4 CPU Loading and Task Info (TDA2xx)

7.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	**HWI**	**1.8**
	SWI	**0.5**
	Total	**9.8%**
DSP1	**HWI**	**0.5**
	SWI	**0.1**
	Total	**20.5%**

7.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
IPU1-0	Stat Collector	Statistics collector	1.9%
	Capture	Capture frames from sensor via VIP port	0.2%
	Display	Display frames via DSS	0.3%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	3.6%
	IPC OUT	To send frame to another processor	0.4%
	IPC IN	To receive frames from another processor	0.2%
DSP1	IPC+ Dup+ Sync+ Merge+ Lane Detect+ Lane Draw	All major processing on DSP1	19.4%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.8%

NOTE: On DSP all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

7.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	
Capture	Output fps	30fps
ALG-Lane Detect (DSP1)	Avg time per frame	4ms
	Output fps	30fps
ALG- Lane Draw (DSP1)	Avg time per frame	0.2ms
	Output fps	30fps
Display	Input fps	30fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The figures are for 720p30fps capture scenario.

7.6 System Memory Usage (TDA2xx)

7.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

7.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	16KB
	HDVPSS Descriptor Mem	1MB	705KB
DSP1	L2	223 KB	22KB
	Local Heap	512 KB	15KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158KB

7.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	DOF
EMIF Read Only	Avg	337 MB/s
	Peak	689 MB/s
EMIF Write Only	Avg	54 MB/s
	Peak	283 MB/s
EMIF Read + Write	Avg	391 MB/s
	Peak	966 MB/s

7.7 Other Benchmarks (TDA2xx)

7.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 2)	Avg	6ms
	Min	5ms
	Max	9ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

7.7.2 Boot Time

PARAMETER	DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059s
main() to Use-case create	0.340 s
Use-case create start to Live preview on display	0.063 s
Total Boot time	**2.462 s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

8 Single Channel Analytics Usecase on TDA2xx

8.1 Overview

This configuration is used to demonstrate the system performance for running multi-algo demo TDA2xx.

8.2 DataFlow

8.2.1 Single channel analytics example

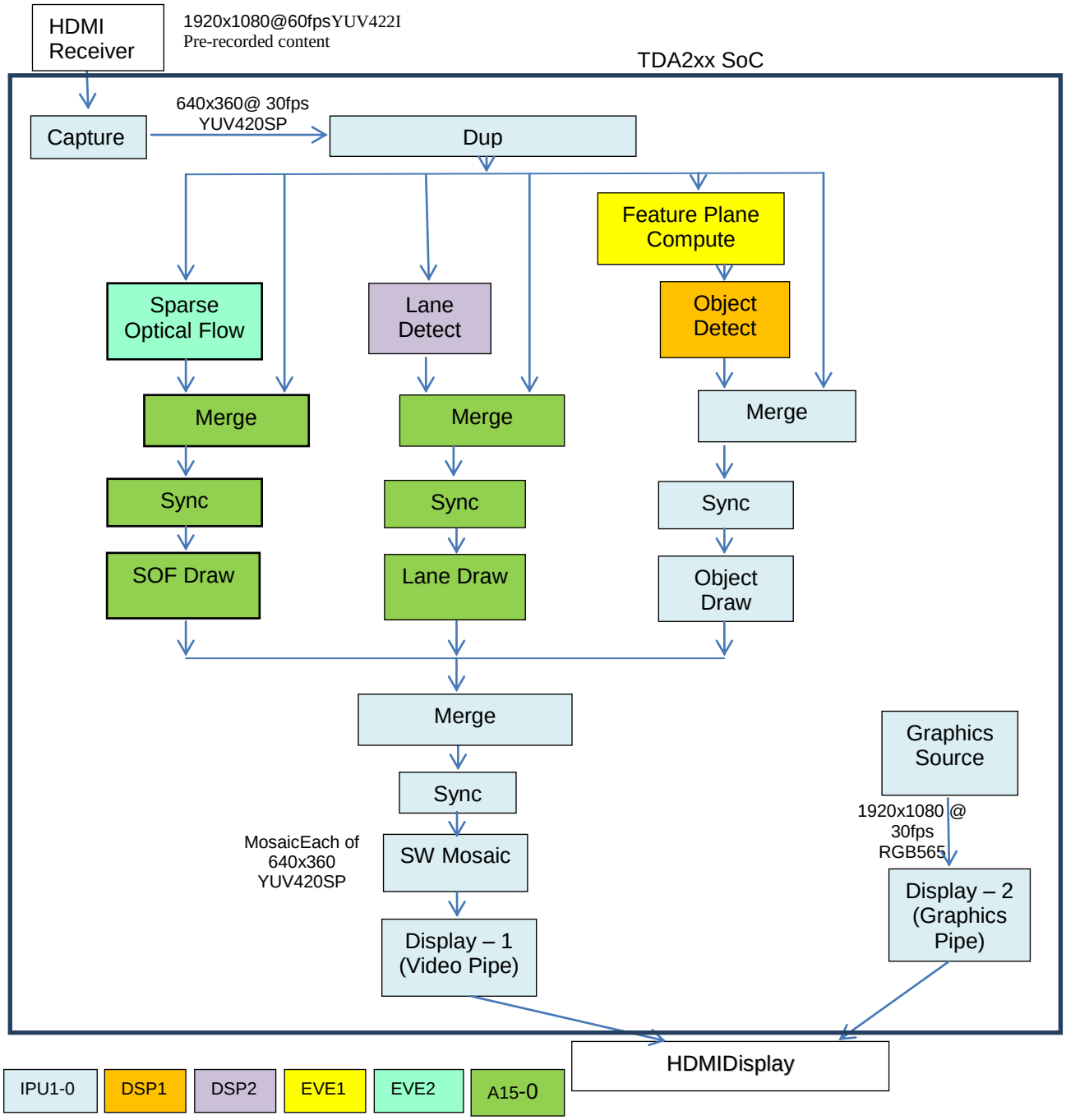
The Capture is HDMI @1080p60. The capture output is set to 640x360. This captured buffer is duplicated by DUP link running on IPU1-0. One output of Dup goes to a sparse optical flow algorithm running on EVE2. This generates metadata output which is subsequently merged along-with the original captured video frames and synced on A15.The output of sync link is sent to SOF draw link running on A15, which draws arrows on the video to depict the optical flow.

Another output of Dup goes to lane detect algorithm running on DSP2. This generates metadata output which is subsequently merged along-with the original captured video frames and synced on A15.The output of sync link is sent to lane draw link running on A15, which draws lines on the video to depict lanes.

Another output of Dup goes to feature plane compute running on EVE1 which generates metadata output buffer. The Object Detect link running on DSP1 utilizes this metadata information to generate co-ordinates where Objects i.e. pedestrians and traffic signs are present in the frame. These co-ordinates are merged and synced with captured frames on IPU1_0. The output of sync link is processed by Object draw algorithm plugin link. If the identified object is a pedestrian then it draws a rectangle around the detected pedestrian. If the identified object is a traffic sign, it finds corresponding bitmap and draws it adjacent to the sign location.

The above three outputs are merged, synced and made into a 3x1 SW mosaic and sent to Display.

The data flow below shows HDMI pre-recorded data fed to chain and HDMI display.



*IPC IN/OUT blocks are left-out to improve readability.

8.3 System Parameters

Refer to section 17.1 for common system parameters.

The benchmarks in this section are computed for HDMI capture (1080p60) of pre-recorded content and HDMI display (1080p60) scenario.

8.4 CPU Loading and Task Info

8.4.1 Total CPU Load

CPU	LOAD TYPE	TDA2XX CPU LOAD
IPU1_0	**HWI**	**3.6**
	SWI	**0.8**
	Total	**22.7 %**
DSP1	**HWI**	**0.3**
	SWI	**0.1**
	Total	**32.8 %**
DSP2	**HWI**	**0.7**
	SWI	**0.1**
	Total	**30.7 %**
EVE1	**HWI**	**1.0**
	SWI	**0.3**
	Total	**74.7 %**
EVE2	**HWI**	**1.0**
	SWI	**0.2**
	Total	**40.5 %**
A15	**HWI**	**0.2**
	SWI	**0.1**
	Total	**11.5**

8.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	TDA2XX CPU LOAD
IPU1-0	Stat Collector	Statistics collector	2.8%
	Capture	Capture frames from sensor via VIP port	0.3 %
	DUP	Duplicate frame to send to display without scaling as well as to VPE to scale	0.3 %
	Sync (2 links)	Sync frames based on timestamp from multiple channels	1.5 %
	Merge (2 links)	Merge frames from SW Mosaic and original capture output	0.6 %
	Display	Display frames via DSS	0.4 %
	ALG – Object Draw	Object detection algorithm	2.9 %

CPU	TASK NAME	TASK DESCRIPTION	TDA2XX CPU LOAD
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	6.2 %
	IPC OUT	To send frame to another processor	1.3 %
	IPC IN	To receive frames from another processor	0.6%
DSP1	IPC + ALG Object Detect	IPC + Object detection algorithm	31.3 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.7 %
DSP2	IPC + ALG Lane Detect	IPC + multiple algorithms	29.1 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.1 %
EVE1	IPC + Feature plane compute	Feature plane compute algorithm + IPC	73.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.6 %
EVE2	IPC + ALG SOF	Sparse Optical Flow algorithm	38.7 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.5 %
A15	ALG SOF Draw	SOF draw algorithm	0.7 %
	ALG Lane Draw	Lane detect draw algorithm	8.0 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	2.3 %

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load calculations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics Load is not accounted for in these measurements.

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

8.5 System Performance

COMPONENT	PARAMETER	TDA2XX
Capture	Output fps	30fps
ALG – SOF (EVE2)	Output fps	30fps
	Avg time per frame	12ms
ALG – SOF Draw (A15)	Output fps	30fps
	Avg time per frame	7ms
ALG – Lane Detect (DSP2)	Output fps	30fps
	Avg time per frame	6ms
ALG – Lane Draw (A15)	Output fps	30fps
	Avg time per frame	0.2ms
Feature Plane Compute (EVE1)	Output fps	30fps
	Avg time per frame	24ms
Object Detect (DSP1)	Output fps	30fps
	Avg time per frame	15ms
Object Draw (IPU1-0)	Output fps	30fps
	Avg time per frame	0.5ms
Display VID 1	Input fps	30fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The figures are for HDMI capture scenario.

8.6 System Memory Usage

8.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage

8.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	TDA2XX MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	20 KB
	HDVPSS Descriptor Mem	1MB	705KB
DSP1	L2	223 KB	223 KB
	Local Heap	512 KB	13 KB
DSP2	L2	223 KB	2 KB
	Local Heap	512 KB	13 KB
EVE1	L2	24 KB	19 KB
	Local Heap	256KB	64 KB
EVE2	L2	24 KB	13 KB
	Local Heap	256 KB	16 KB
A15	Local Heap	4096 KB	56 KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158KB

8.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	TDA2XX
EMIF Read Only	Avg	1077 MB/s
	Peak	2309 MB/s
EMIF Write Only	Avg	390 MB/s
	Peak	1281 MB/s
EMIF Read + Write	Avg	1465 MB/s
	Peak	2886 MB/s

8.7 Other Benchmarks

8.7.1 Processing Latency

		TDA2XX LATENCY
Capture to Display Latency (Display VID 2)	Avg	40ms
	Min	32ms
	Max	49ms

NOTE:

- This latency is as measured inside the system by software.

- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency
-

8.7.2 Boot Time

PARAMETER	TDA2XX DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s
main() to Use-case create	0.340 s
Use-case create start to Live preview on display	0.575 s
Total Boot time	**2.974 s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

9 Single Channel Analytics Usecase on TDA3xx

9.1 Overview

This configuration is used to demonstrate the system performance of running multiple algorithms on TDA3xx.

9.2 DataFlow

9.2.1 Single channel analytics example

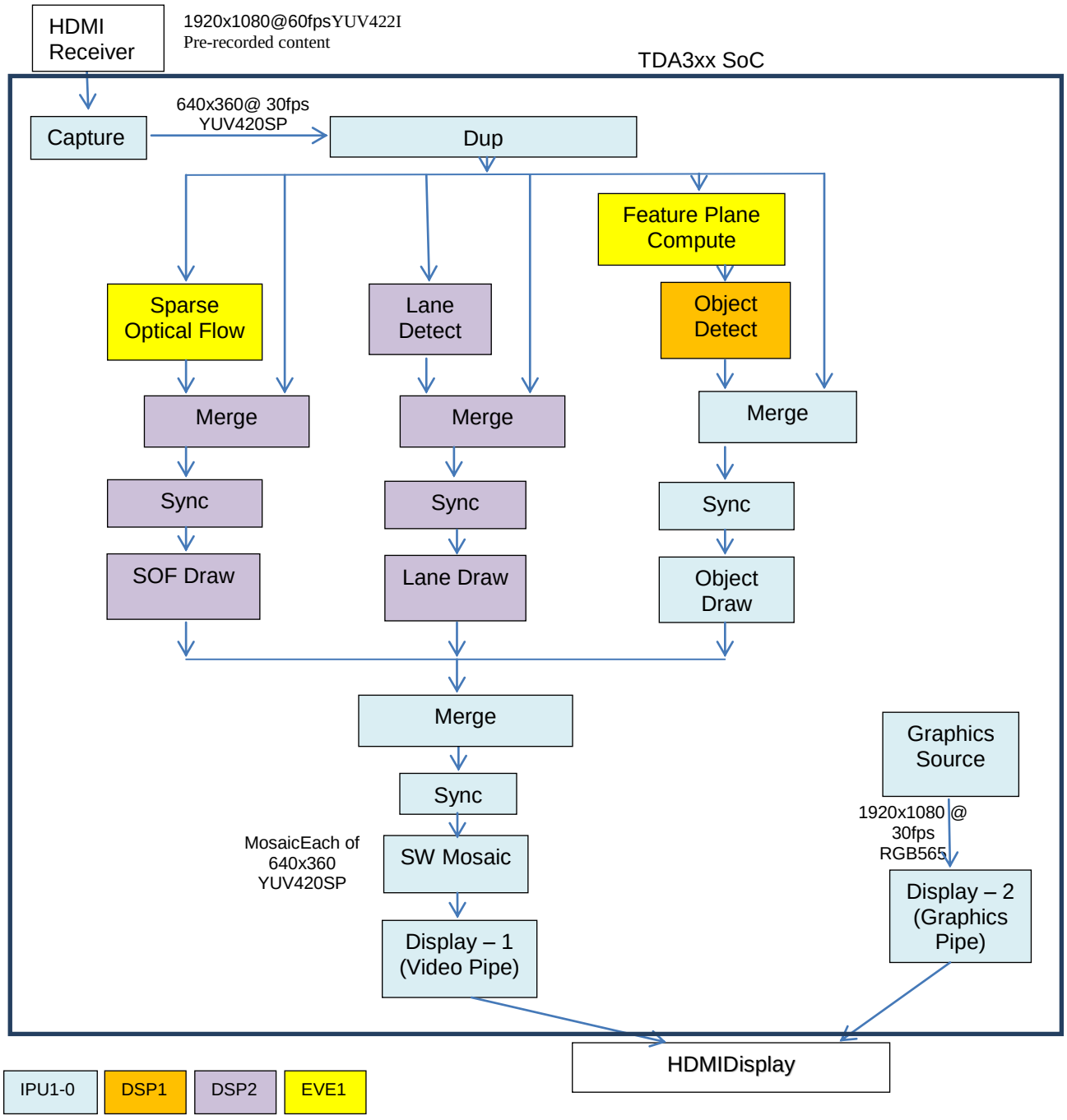
The Capture is HDMI @1080p60. The capture output is set to 640x360. This captured buffer is duplicated by DUP link running on IPU1-0. One output of Dup goes to a sparse optical flow algorithm running on EVE1. This generates metadata output which is subsequently merged along-with the original captured video frames and synced on DSP2.The output of sync link is sent to SOF draw link running on DSP2, which draws arrows on the video to depict the optical flow.

Another output of Dup goes to lane detect algorithm running on DSP2. This generates metadata output which is subsequently merged along-with the original captured video frames and synced on DSP2.The output of sync link is sent to lane draw link running on DSP2, which draws lines on the video to depict lanes.

Another output of Dup goes to feature plane compute running on EVE1 which generates metadata output buffer. The Object Detect link running on DSP1 utilizes this metadata information to generate co-ordinates where Objects i.e. pedestrians and traffic signs are present in the frame. These co-ordinates are merged and synced with captured frames on IPU1_0. The output of sync link is processed by Object draw algorithm plugin link. If the identified object is a pedestrian then it draws a rectangle around the detected pedestrian. If the identified object is a traffic sign, it finds corresponding bitmap and draws it adjacent to the sign location.

The above three outputs are merged, synced and made into a 3x1 SW mosaic and sent to Display.

The data flow below shows HDMI pre-recorded data fed to chain and HDMI display.



**IPC IN/OUT blocks are left-out to improve readability.

9.3 System Parameters

Refer to section 20.1for common system parameters.

The benchmarks in this section are computed for HDMI capture (1080p60) of pre-recorded content and HDMI display (1080p60) scenario.

9.4 CPU Loading and Task Info

9.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD (%)
IPU1_0	**HWI**	**2.9**
	SWI	**0.6**
	Total	**14.8**
DSP1	**HWI**	**0.3**
	SWI	**0.1**
	Total	**32.6 %**
DSP2	**HWI**	**0.9**
	SWI	**0.1**
	Total	**64.7**
EVE1	**HWI**	**1.1**
	SWI	**0.2**
	Total	**98.2**

9.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	TDA3XX CPU LOAD
IPU1-0	Stat Collector	Statistics collector	Na
	Capture	Capture frames from sensor via VIP port	0.2 %
	DUP	Duplicate frame to send to display without scaling as well as to VPE to scale	0.2 %
	Sync (2 links)	Sync frames based on timestamp from multiple channels	1.0 %
	Merge (2 links)	Merge frames from SW Mosaic and original capture output	0.4 %
	Display (2 links)	Display scaled frames via DSS	0.6 %
	ALG – Object Draw	Object detection algorithm	2.1 %
	Alg- SW Mosaic		0.6 %

CPU	TASK NAME	TASK DESCRIPTION	TDA3XX CPU LOAD
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	5.3 %
	IPC OUT (5 links)	To send frame to another processor	1.1 %
	IPC IN (3 links)	To receive frames from another processor	0.5 %
DSP1	IPC + ALG Object Detect	IPC + Object detection algorithm	32.1 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.6 %
DSP2	IPC + ALG Lane Detect + Lane Draw + SOF draw	IPC + multiple algorithms	63.7 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.2 %
EVE1	IPC + Feature plane compute	Feature plane compute algorithm + IPC	96.4 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.5 %

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load calculations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements.

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

9.5 System Performance

COMPONENT	PARAMETER	TDA3XX
Capture	Output fps	28.2fps
ALG – SOF (EVE1)	Output fps	28.75fps
	Avg time per frame	10.0ms
ALG – SOF Draw (DSP2)	Output fps	22fps
	Avg time per frame	0.7ms
ALG – Lane Detect (DSP2)	Output fps	22fps
	Avg time per frame	7ms
ALG – Lane Draw (DSP2)	Output fps	22fps
	Avg time per frame	0.7ms
Feature Plane Compute (EVE1)	Output fps	22fps
	Avg time per frame	33ms
Object Detect (DSP1)	Output fps	22fps
	Avg time per frame	21ms
Object Draw (IPU1-0)	Output fps	22fps
	Avg time per frame	0.3ms
Display VID 1	Input fps	22fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The figures are for HDMI capture scenario.

9.6 System Memory Usage

9.6.1 Code/Data Memory Usage

Refer section20.2for common Code/Data Memory usage

9.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	TDA3XX MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	14 KB
	HDVPSS Descriptor Mem	256 KB	106KB
DSP1	L2	223KB	223KB
	Local Heap	512 KB	10 KB
DSP2	L2	223 KB	22 KB
	Local Heap	512 KB	13 KB
EVE1	L2	24 KB	19 KB
	Local Heap	256 KB	78 KB
Shared Memory	SR0	12 MB	2 MB
	Frame Buffer (SR1)	256 MB	256 MB
	Remote Log Buffer	160KB	158KB

9.6.3 DDR Bandwidth

These numbers are currently not available for TDA3xx

9.7 Other Benchmarks

9.7.1 Processing Latency

		TDA3XX LATENCY
Capture to Display Latency (Display VID 2)	Avg	65ms
	Min	36ms
	Max	152ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency
-

9.7.2 Boot Time

PARAMETER	TDA3XX DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	0. 908 s
main() to Use-case create	0.222 s
Use-case create start to Live preview on display	0.689 s
Total Boot time	**2.011s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

10 Multi-channel LVDS Surround view + Object DetectUse case on TDA2xx

10.1 Overview

This use case consists of capture from multiple OV10635 camera's. The camera's are connected to TDA2xx via FPD link with serializer and de-serializers in between. This configuration demonstrates a Surround View on 4 Channels input and Analytics on single channel in parallel. Only HDMI display is supported by this use-case.

10.2 Data Flow

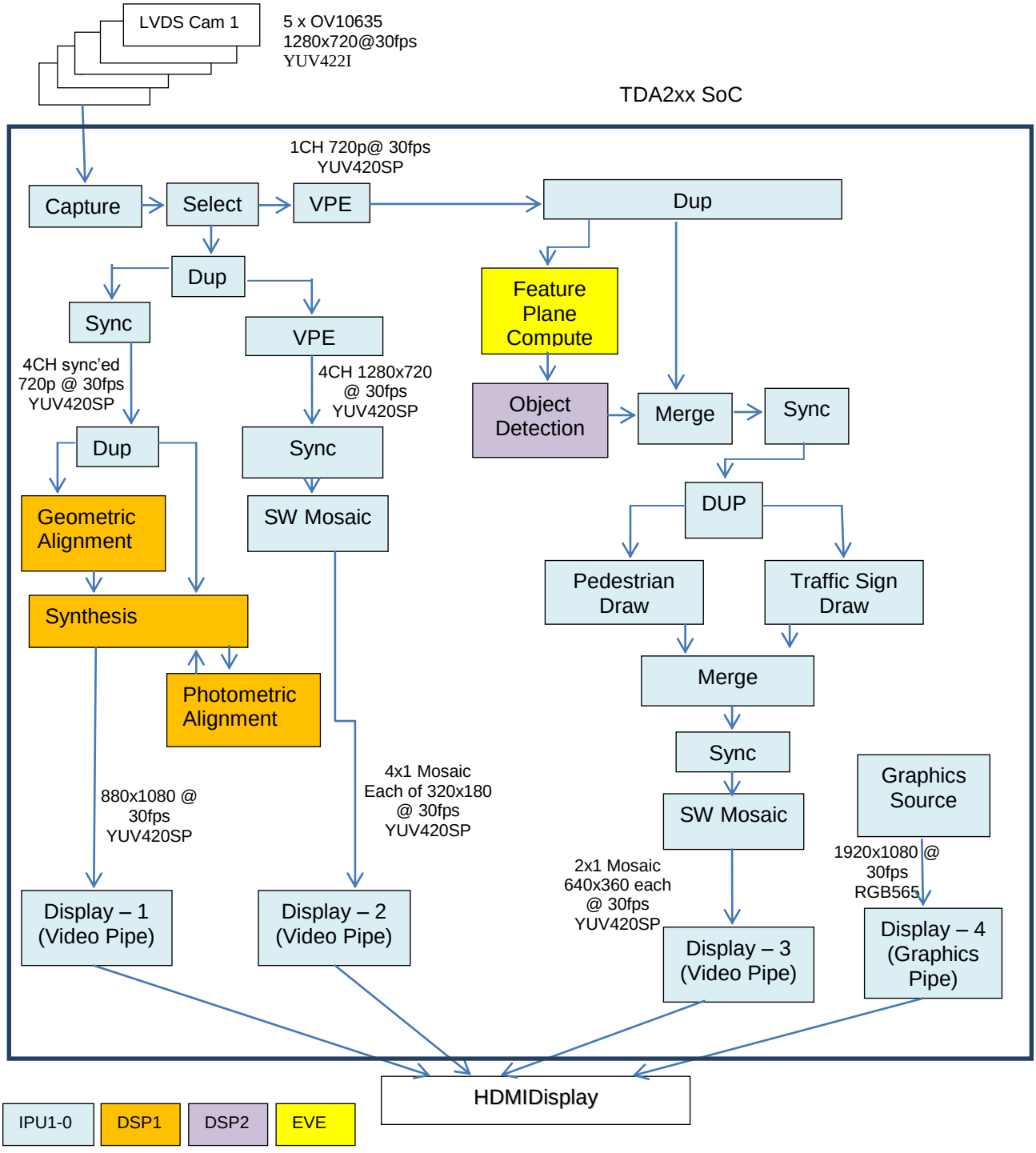
10.2.1 5CH LVDS capture, Surround View demonstration

In this configuration we capture 5 Channel video from 5 OV1063x sensors @ 720p 30fps.

Of these, 1 Channel data is dup'ed into 2 outputs using DUP link. On this data Feature plane compute (EVE1) and Object detect (DSP2) algorithms are applied. Object detect output is merged with original channel data. This merged data is sync'ed and dup'ed, and objects i.e. Pedestrians and traffic signs are respectively indicated using two object Draw alg plugins (IPU1_0). These are finally Merged, Synced and displayed onto HDMI as a 2x1 Mosaic.

The other 4 Channels data is sent to Sync Link where these captured frames are synced based on time stamps. After syncing we get a "group of sync'ed frames", one frame from each camera. This "group of sync'ed frames" is passed on to algorithm links of Geometric alignment and synthesis link. Geometric alignment link provides look up table for geometric alignment among multiple views. Further Photometric alignment provides look up table for pixel transformations to compensate for the difference in lighting among different cameras. Based on these two look up tables, the Synthesis stage generates surround view which is shown on the display. The original image from these 4 Channels is downscaled using a VPE and displayed as a 4x1 mosaic of 320x180 @30fps.

Please note in dataflow diagram below IPC IN/OUT blocks are left-out to improve readability. Please assume these whenever CPU changes in the flow.



**IPC IN/OUT blocks are left-out to improve readability.

IPC IN/OUT blocks are left-out to improve readability

10.3 System Parameters (TDA2xx)

Refer section 17.1 for common system parameters.

The parameters in this section are computed for HDMI display.

10.4 CPU Loading and Task Info (TDA2xx)

10.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	HWI	6.4 %
	SWI	1.6 %
	Total	40.0%
DSP1	HWI	0.9 %
	SWI	0.1 %
	Total	88.6%
DSP2	HWI	0.6 %
	SWI	0.1 %
	Total	51.2%
EVE1	HWI	1.0 %
	SWI	0.2 %
	Total	75.8 %

10.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (SRV 5CH)
IPU1-0	Stat Collector	Statistics collector	3.4 %
	Capture	Capture frames from sensor via VIP port	1.0%
	Display (4 links)	Display frames/graphics via DSS	2.2%
	Object Draw (2 links)	Pedestrian and TSR draw	18.0 %
	VPE (2 links)	Scale frames	4.4%
	SYNC (4 links)	Sync frames based on timestamp from multiple channels	3.1%
	DUP (4 links)	Duplicate frame to send to display without scaling as well as to VPE to scale	1.0%
	MERGE (2 links)	Merge frames from SW Mosaic and original capture output	0.5%
	Select	Selects specific channel data from i/p queue	0.8%
	SW Mosaic (2 links)	Composite synced frames from multiple channels to form a single composite frame	1.3%

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (SRV 5CH)
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	10.0%
	IPC OUT (4 links)	To send frame to another processor	1.5%
	IPC IN (2 links)	To receive frames from another processor	0.4%
DSP1	IPC +GAlign+PAlign+Synthesis +	All DSP1 Processing algorithms	89.4
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.4%
DSP2	IPC + ALG Object Detect	IPC+Object detect algorithm	50.3%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.0%
EVE1	ALG Feature Plane COmpute+IPC	Feature Plane compute algorithm + IPC	84.1%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.4%

*NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

10.5 System Performance (TDA2xx)

COMPONENT	PARAMETER	SRV 5CH FOR *Capture to Display1 path ONLY*
Capture	Output fps	30fps on each Channels
Feature Plane Compute (EVE1)	Output fps	30fps
	Avg time per frame	27ms
Object Detect (DSP2)	Output fps	30fps
	Avg time per frame	20ms
Object Draw (IPU1-0)	Output fps	30fps
	Avg time per frame	3.8ms
ALG Synthesis (DSP1)	Output fps	30fps
	Avg time per frame	28ms
ALG – Photometric Align(DSP1)	Output fps	30fps
	Avg time per frame	0.9ms
Display VID 1	Input fps	30fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The above figures are for HDMI display.

10.6 System Memory Usage (TDA2xx)

10.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage.

10.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	15 KB
	HDVPSS Descriptor Mem	1MB	705KB
DSP1	L2	223 KB	128 KB
	Local Heap	512 KB	9 KB
DSP2	L2	223 KB	223 KB
	Local Heap	512 KB	16 KB
EVE1	L2	24 KB	19 KB
	Local Heap	256 KB	64 KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158KB

10.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	
EMIF1 Read Only	Avg	1400 MB/s
	Peak	2126 MB/s
EMIF1 Write Only	Avg	472 MB/s
	Peak	765 MB/s
EMIF1 Read + Write	Avg	1871 MB/s
	Peak	2714 MB/s

10.7 Other Benchmarks (TDA2xx)

10.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 1)(SV)	Avg	157 ms
	Min	81 ms
	Max	275 ms
Capture to Display Latency (Display VID 3)(FC ANalytics)	Avg	71 ms
	Min	56 ms
	Max	112 ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

10.7.2 Boot Time

PARAMETER	VALUE
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059s
main() to Use-case create	0.340 s
Use-case create start to Live preview on display	2.546s
Total Boot time	4.945s

QSPI Boot time measurement done with TDA2xx ES1.0 samples.

GUI and Sensor initialization time not accounted for.

11 Multi-channel LVDS Surround view Use case on TDA2Ex

11.1 Overview

This use case consists of capture from multiple OV10635 camera's. The camera's are connected to TDA2Ex via FPD link with serializer and de-serializers in between. This configuration demonstrates a Surround View on 4 Channels input. Only HDMI display is supported by this use-case.

11.2 Data Flow

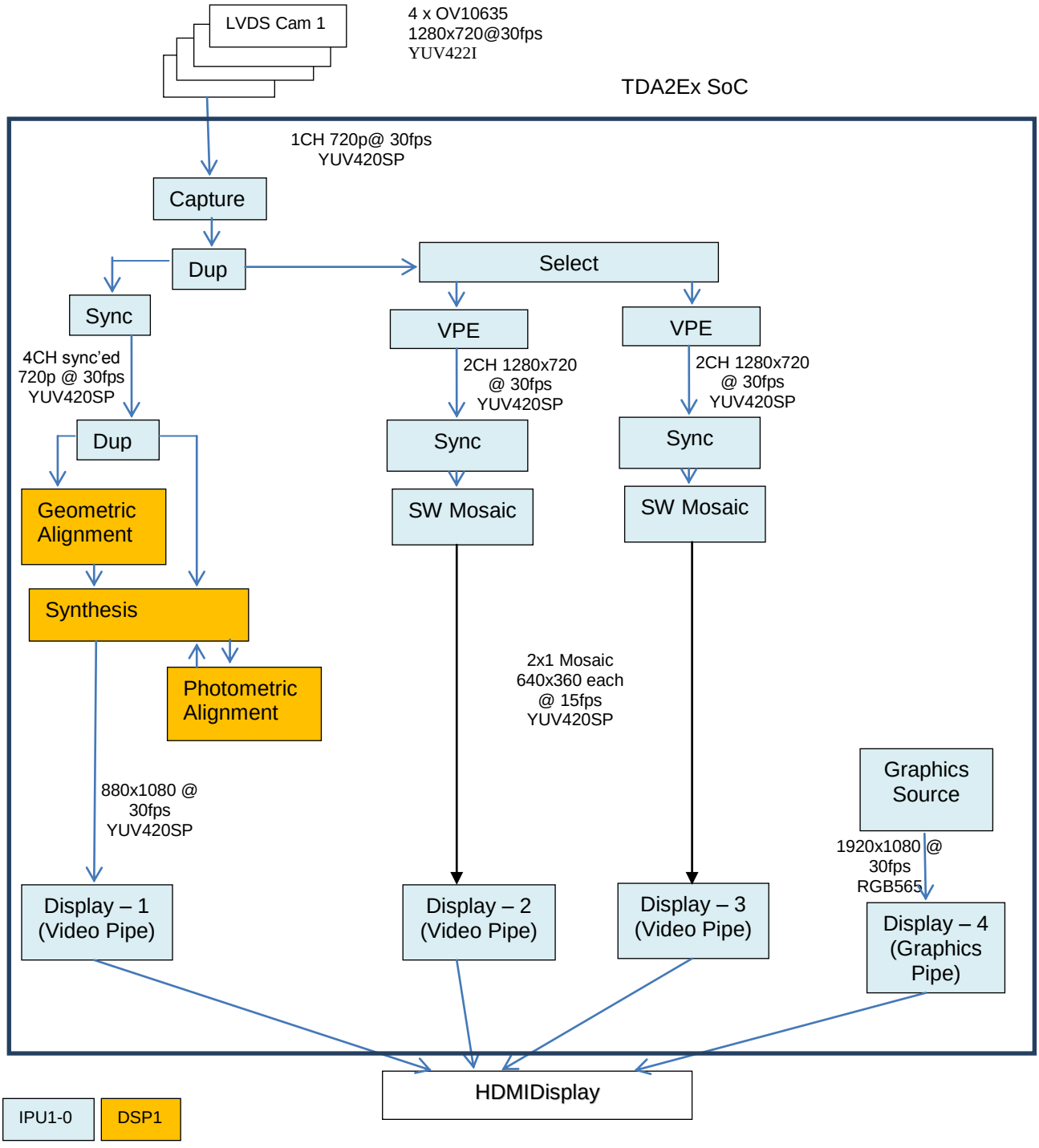
11.2.1 4CH LVDS capture, Surround View demonstration

In this configuration we capture 4 Channel video from 4 OV1063x sensors @ 720p 30fps.

This 4 Channel data is duped. Using Select, VPE, Sync and Mosaic links the original data is downscaled and displayed as two 2x1 Mosaics.

The 4 Channels data via second path on DUP is sent to Sync Link where these captured frames are synced based on time stamps. After syncing we get a "group of sync'ed frames", one frame from each camera. This "group of sync'ed frames" areDup'ed. One set is passed on to algorithm links of Geometric alignment and synthesis link. Geometric alignment link provides look up table for geometric alignment among multiple views. Further Photometric alignment provides look up table for pixel transformations to compensate for the difference in lighting among different cameras. Based on these two look up tables and original 4 Ch Data, the Synthesis stage generates surround view which is shown on the display.

Please note in dataflow diagram below IPC IN/OUT blocks are left-out to improve readability. Please assume these whenever CPU changes in the flow.



**IPC IN/OUT blocks are left-out to improve readability.

11.3 System Parameters (TDA2Ex)

Refer section 19.1for common system parameters.

The parameters in this section are computed for HDMI display scenario and **Dual EMIF.

11.4 CPU Loading and Task Info (TDA2Ex)

11.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD (%)
IPU1_0	HWI	4.5
	SWI	1.4
	Total	20.0
DSP1	HWI	0. 6
	SWI	0.1
	Total	87.0

11.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)
	Capture	Capture frames from sensor via VIP port	0.7
	Display (4 links)	Display frames/graphics via DSS	1.4
	VPE (2 links)	Scale frames	3.3
	SYNC (3 links)	Sync frames based on timestamp from multiple channels	2.1
	DUP (2 links)	Duplicate frame to send to display without scaling as well as to VPE to scale	0.8
	Select	Selects specific channel data from i/p queue	0.4
	SW Mosaic (2 links)	Composite synced frames from multiple channels to form a single composite frame	1.0
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	1.7
	IPC OUT (2 links)	To send frame to another processor	0.9
	IPC IN (2 links)	To receive frames from another processor	0.2
DSP1	IPC +GAlign+PAlign+Synthesis +	All DSP1 Processing algorithms	86.2

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.8

*NOTE: On DSP and similarly on EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

11.5 System Performance (TDA2Ex)

COMPONENT	PARAMETER	SRV 5CH FOR *Capture to Display1 path ONLY*
Capture	Output fps	26fps each on 4 Channels for SV
ALG Synthesis (DSP1)	Output fps	26fps
	Avg time per frame	26ms
ALG – Photometric Align(DSP1)	Output fps	26fps
	Avg time per frame	0.89ms
Display VID 1	Input fps	26fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer
NOTE: The above figures are for HDMI display.

11.6 System Memory Usage (TDA2Ex)

11.6.1 Code/Data Memory Usage

Refer section 19.2for common Code/Data Memory usage.

11.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	27 KB
	HDVPSS Descriptor Mem	2MB	1 MB
DSP1	L2	224 KB	128 KB
	Local Heap	512 KB	10 KB
Shared Memory	SR0	12 MB	1 MB
	Frame Buffer (SR1)	256 MB	256 MB
	SR2 OCMC	511 KB	0 KB
	Remote Log Buffer	160 KB	158 KB

11.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	
EMIF1 Read Only	Avg	981 MB/s
	Peak	1434 MB/s
EMIF1 Write Only	Avg	306 MB/s
	Peak	450MB/s
EMIF1 Read + Write	Avg	1287 MB/s
	Peak	1865 MB/s

11.7 Other Benchmarks (TDA2Ex)

11.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 1)(SV)	Avg	10 ms
	Min	9 ms
	Max	22 ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

11.7.2 Boot Time

PARAMETER	VALUE
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	0.775 s
main() to Use-case create	0.849 s
Use-case create start to Live preview on display	2.428 s
Total Boot time	4.050 s

QSPI Boot time measurement done with TDA2Ex ES1.0 samples.

GUI and Sensor initialization time not accounted for.

12 Multi-channel AVB Surround view Use case on TDA2xx & TDA2ex

12.1 Overview

This use case consists of AVB capture (Ethernet based input).The received data is in MJPEG format and decoded. This configuration demonstrates a Surround View on 4 Channels input and Edge Detect on single channel in parallel. Only HDMI display is supported by this use-case.

12.2 Data Flow

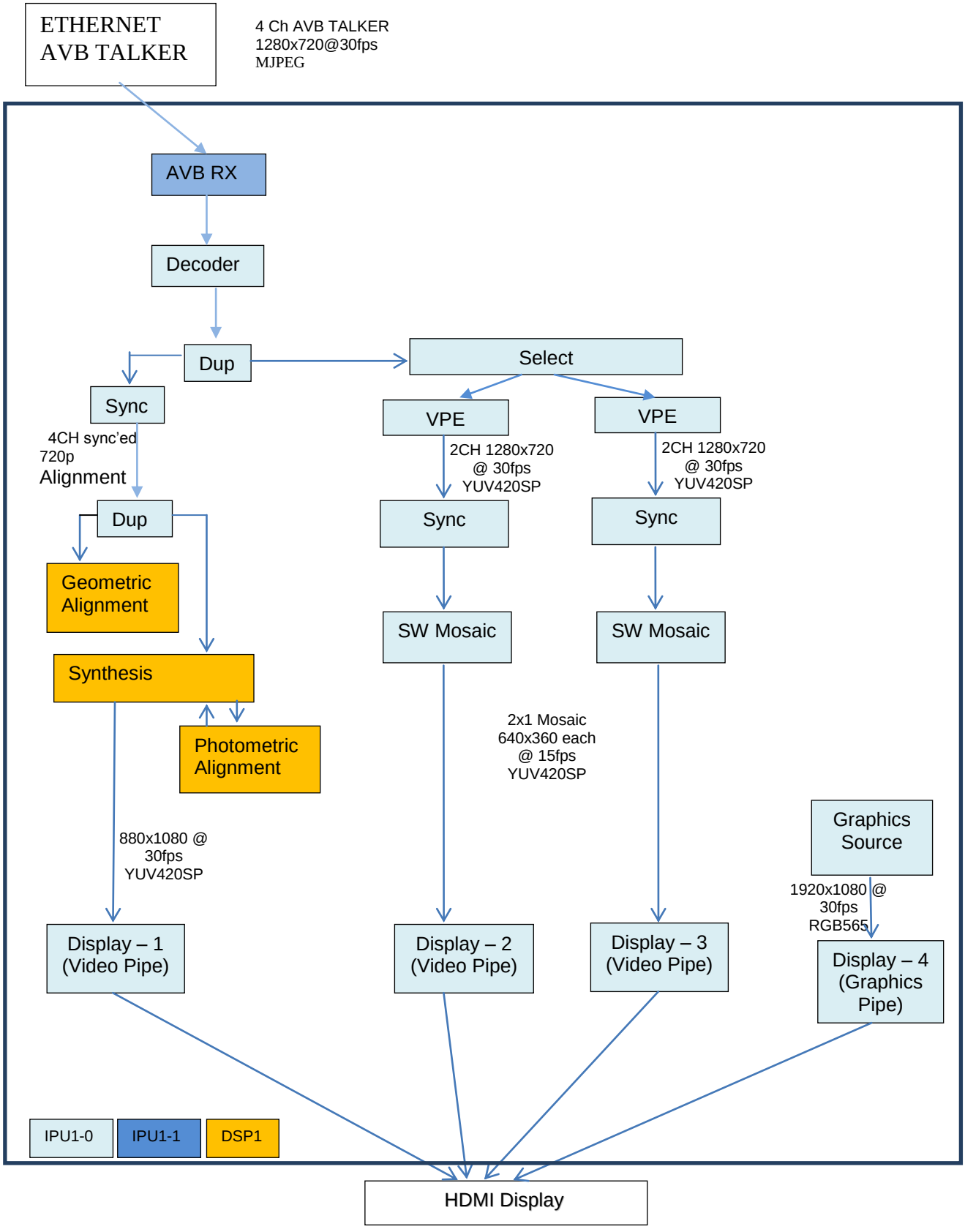
12.2.1 Configuration 10: 4CH AVB capture, Surround View demonstration

In this configuration we receive 4 Channel video from AVB Talker @ 720p 30fps per channel. These streams are decoded using MJPEG decoder. The original image from this channel is then Merged, Synced and displayed onto HDMI as a 2x1 Mosaic.

This 4 Channel data is duped. Using Select, VPE, Sync and Mosaic links the original data is downscaled and displayed as two 2x1 Mosaics.

The 4 Channels data via second path on DUP is sent to Sync Link where these captured frames are synced based on time stamps. After syncing we get a "group of sync'ed frames", one frame from each camera. This "group of sync'ed frames" are Dup'ed. One set is passed on to algorithm links of Geometric alignment and synthesis link. Geometric alignment link provides look up table for geometric alignment among multiple views. Further Photometric alignment provides look up table for pixel transformations to compensate for the difference in lighting among different cameras. Based on these two look up tables and original 4 Ch Data, the Synthesis stage generates surround view which is shown on the display.

Please note in dataflow diagram below IPC IN/OUT blocks are left-out to improve readability. Please assume these whenever CPU changes in the flow.


ETHERNET AVB TALKER
4 Ch AVB TALKER
1280x720@30fps
MJPEG
AVB RX
Decoder
Dup
Select
Sync
4CH sync'ed 720p Alignment
VPE
VPE
2CH 1280x720 @ 30fps YUV420SP
2CH 1280x720 @ 30fps YUV420SP
Dup
Sync
Sync
Geometric Alignment
SW Mosaic
SW Mosaic
Synthesis
Photometric Alignment
2x1 Mosaic 640x360 each @ 15fps YUV420SP
880x1080 @ 30fps YUV420SP
Graphics Source
1920x1080 @ 30fps RGB565
Display – 1 (Video Pipe)
Display – 2 (Video Pipe)
Display – 3 (Video Pipe)
Display – 4 (Graphics Pipe)
IPU1-0
IPU1-1
DSP1
HDMI Display

12.3 System Parameters (TDA2xx)

Refer section 17.1 for common system parameters.

The parameters in this section are computed for HDMI display scenario.

12.4 CPU Loading and Task Info(TDA2xx)

12.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD TDA2XX (%)	CPU LOAD TDA2EX (%)
IPU1-0	HWI	6.2	4.8
	SWI	2.4	1.5
	Total	46.4	39.8
IPU1-1	HWI	9.4	8.0
	SWI	1.0	0.7
	Total	69	58.9
DSP1	HWI	0.8	0.6
	SWI	0.1	0.1
	Total	87.9%	88.3

12.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (SRV 4CH) TDA2XX	CPU LOAD (SRV 4CH) TDA2EX
IPU1-0	Stat Collector	Statistics collector	2.3%	2.6%
	Decoder	Decoder the mjpeg frame	12.8%	12.8%
	Display (4 links)	Display frames	2.8%	2.3%
	Sync (3 links)	Sync frames based on timestamp from multiple channels	3.4%	3.9%
	Select	Selects an input channel from queue	0.8%	0.8%
	VPE (two links)	Scale frames	7.3%	5.8%
	DUP (2 links)	Duplicate frame to send to display without scaling as well as to VPE to scale	1.2%	1.4%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	2.3%	8.2%
IPU1-1	AVB	Receives frames from network		57%

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (SRV 4CH) TDA2XX	CPU LOAD (SRV 4CH) TDA2EX
	IPC OUT (2 links)	To send frame to another processor	1.8%	1. 8%
	IPC IN (2 links)	To receive frames from another processor	1.1%	0.9%
DSP1	IPC + ALG GALIGN + ALG PALIGN	IPC+ Alg processing	3.0%	90%
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.3%	0.9%

*NOTE: On EVE all links run in a single thread.

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load calculations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements.

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

12.5 System Performance (TDA2xx and TDA2ex)

COMPONENT	PARAMETER	SRV 4CH FOR *Capture to Display1 path ONLY TDA2xx*	SRV 4CH FOR *Capture to Display1 path ONLY TDA2ex*
AVB (IPU1-1)	Output FPS	160fps over 4 channels	160fps over 4 channels
	Data Rate	360 Mbps	360 Mbps
Decoder	Output FPS	160fps over 4 channels	160fps over 4 channels
	Avg time per frame	27ms	32ms
ALG	Output fps	30fps	32fps

Synthesis (DSP1)	Avg time per frame	27ms	27ms
ALG – Photometric Align(DSP1)	Output fps	30fps	30fps
	Avg time per frame	0.9ms	0.9ms
Display	Input FPS	30fps	30fps
	VENC FPS	60fps	60fps

NOTE: Since a AVB talker is used instead of live camera feed the fps on each channel is not fixed
NOTE: Only numbers for the AVB Capture to Display 1 path i.e. path for Surround view is shown above
NOTE: FPS numbers are rounded off to nearest integer.
NOTE: The above figures are for HDMI display.

12.6 System Memory Usage (TDA2xx & TDA2ex)

12.6.1 Code/Data Memory Usage

Refer section 17.2for common Code/Data Memory usage.

12.6.2 Heap Memory Usage TDA2xx

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	37 KB
	HDVPSS Descriptor Mem	2MB	1 MB
IPU1-1	Local Heap	256 KB	44 KB
DSP1	L2	224 KB	128 KB
	Local Heap	512 KB	14 KB
Shared Memory	SR0	12 MB	4 MB
	Frame Buffer (SR1)	256 MB	256 MB
	SR2 OCMC	1023 KB	0 KB
	Remote Log Buffer	160 KB	158 KB

12.6.3 DDR Bandwidth TDA2xx

PARAMETER	BANDWIDTH	
EMIF Read Only	Avg	1230 MB/s
	Peak	2000 MB/s
EMIF Write Only	Avg	463 MB/s
	Peak	749 MB/s
EMIF Read + Write	Avg	1691 MB/s
	Peak	2748 MB/s

12.6.4 Heap Memory Usage TDA2ex

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	16 KB
	HDVPSS Descriptor Mem	2MB	1 MB
IPU1-1	Local Heap	256 KB	28 KB
DSP1	L2	224 KB	128 KB
	Local Heap	512 KB	9 KB
Shared Memory	SR0	12 MB	4 MB
	Frame Buffer (SR1)	255 MB	129 MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160 KB	158 KB

12.6.5 DDR Bandwidth TDA2ex

PARAMETER	BANDWIDTH	
EMIF Read Only	Avg	1148 MB/s
	Peak	1776 MB/s
EMIF Write Only	Avg	469 MB/s
	Peak	773 MB/s
EMIF Read + Write	Avg	1617 MB/s
	Peak	2446 MB/s

12.7 Other Benchmarks (TDA2xx)

12.7.1 Processing Latency

This benchmark has not been measured for this configuration due to a known issue in Vision SDK stack where the latency figures are not initialized correctly.

12.7.2 Boot Time

PARAMETER	VALUE
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059s
main() to Use-case create	1.30 s
Use-case create start to Live preview on display	2.23s (Talker should be started immediately)
Total Boot time	5.592s

QSPI Boot time measurement done with TDA2xx ES1.0 samples.

GUI and Sensor initialization time not accounted for.

13 ISS Capture Display Use case on TDA3xx

13.1 Overview

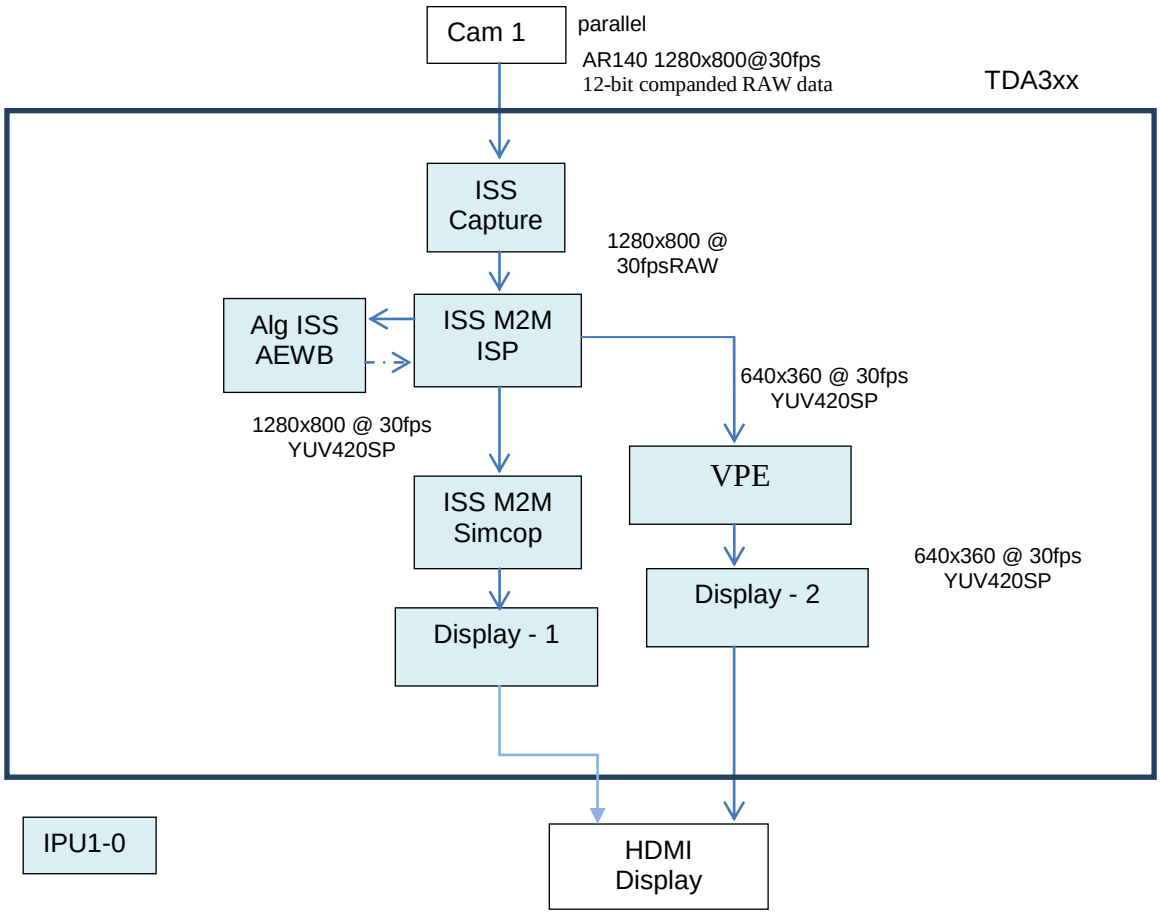
This configuration is used to demonstrate the ISS on TDA3xx.

13.2 DataFlow

13.2.1 ISS Capture Display example

In the ISS use-case, raw bayer data is captured from image sensor, like OV10640 or AR0140, by the ISS capture link. For OV10640 sensor capture is at 1280x720, whereas for AR140 it is at 1280x800. The sensor interface can be CSI2 or parallel. The bits/pixel of the raw data depends on the sensor and sensor mode being used, e.g. with AR140, 12-bit companded mode is used. The raw bayer data is fed to M2M (memory-to-memory) ISP (Image Signal Processor) link to convert it to YUV data. The ISP can operate in 2-pass WDR mode or 1-pass linear mode depending on sensor mode. There are two outputs from M2M ISP – Resizer A output of 1280x800 given to M2M SIMCOP, and Resizer B output of 640x360 sent to HDMI display. The M2M SIMCOP which performs optional LDC (Lens Distortion Correction) and/or VTNF (Video Temporal Noise Filter). The output of M2M SIMCOP is semt to HDMI display. The M2M ISP also outputs statistical data for the AE/AWB algorithm plugin. The AE/AWB algorithm feedback is applied to the M2M ISP link and/or sensor.

The data flow below shows HDMI display.



13.3 System Parameters (TDA3xx)

Refer to section 20.1 for common system parameters.

The benchmarks in this section are computed for AR0140 Parallel sensor and HDMI display (1080p60) scenario.

13.4 CPU Loading and Task Info (TDA3xx)

13.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD
IPU1_0	**HWI**	**2.5 %**
	SWI	**0.5 %**
	Total	**21.2%**

13.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD
IPU1_0	ISSCapture	ISS Capture	0.1 %
	ISSM2MISP	ISS memory to memory ISP	1.7 %
	ISSM2MSIM COP	ISS memory to memory LDC+VTNF	0.7 %
	Display	Display via DSS	0.4 %
	Alg ISS AEWB	ISS Auto White Balance	6.9 %
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	4.3 %

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load caclulations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics load is not accounted for in these measurements.

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log

- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

13.5 System Performance (TDA3xx)

COMPONENT	PARAMETER	
ISS Capture	Output fps	30fps
ISS M2M ISP	Input fps	30fps
	Output fps	30fps
	Avg time per frame	12ms
ISS M2M SIMCOP	Input fps	30fps
	Output fps	30fps
	Avg time per frame	11ms
Algorithm AEWB	Input fps	30fps
	Output fps	30fps
	Avg time per frame	2ms
Display	Input fps	30fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer

13.6 System Memory Usage (TDA3xx)

13.6.1 Code/Data Memory Usage

Refer section 20.2for common Code/Data Memory usage

13.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	10KB
	HDVPSS Descriptor Mem	256KB	106KB
Shared Memory	SR0	8KB	8KB
	Frame Buffer (SR1)	256MB	256MB
	Remote Log Buffer	160KB	158 KB

13.7 Other Benchmarks (TDA3xx)

13.7.1 Processing Latency

		LATENCY
Capture to Display Latency (Display VID 1)	Avg	12 ms
	Min	11 ms
	Max	14 ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency

13.7.2 Boot Time

PARAMETER	DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	0.952s
main() to Use-case create	0.222 s
Use-case create start to Live preview on display	0.581 s
Total Boot time	**1.7555 s**

QSPI Boot time measurement done with TDA2xx ES1.0 samples.

GUI and Sensor initialization time not accounted for.

14 Fast boot ISS Capture + Object Detect + Display

14.1 Over view

This usecase demonstrates fast boot feature that can be applied for any usecase in vision_sdk. This section mentions performance numbers achieved for fast boot usecase from the boot time split perspective.

14.2 Dataflow

The usecase dataflow same as the object detect data flow motioned in section 6.2.1

More details about boot time optimization techniques can be found in VisionSDK_DevelopmentGuide.pdf

14.3 Usecase configuration for boot timer measurement

Please note following build configuration parameters for boot time measurement.

```
#
# Platform config,
#   VSDK_BOARD_TYPE=TDA3XX_EVM [options: TDA2XX_EVM TDA3XX_EVM TDA2XX_MC TDA2EX_EVM]
# PLATFORM=tda3xx-evm
# DDR_MEM=DDR_MEM_512M [options: DDR_MEM_64M DDR_MEM_512M]
#
# Build config,
# BUILD_OS=Linux [options: Windows_NT Linux]
# A15_TARGET_OS=Bios [options: Bios Linux]
# PROFILE=release [options: debug release]
# BUILD_DEPENDANCY_ALWAYS=yes
# BUILD_ALGORITHMS=no
#
# CPU config,
# PROC_IPU1_0_INCLUDE=yes
# PROC_IPU1_1_INCLUDE=no
# PROC_DSP1_INCLUDE=yes
# PROC_DSP2_INCLUDE=no
# PROC_EVE1_INCLUDE=yes
# PROC_EVE2_INCLUDE=no
# PROC_EVE3_INCLUDE=no
# PROC_EVE4_INCLUDE=no
# PROC_A15_0_INCLUDE=no
#
# Module config,
# NDK_PROC_TO_USE=none
# AVBRX_INCLUDE=no
# DCAN_INCLUDE=no
```

```
# IVAHD_INCLUDE=no
# VPE_INCLUDE=no
# ISS_INCLUDE=yes
# DSS_INCLUDE=yes
# HCF_INCLUDE=no
# CRC_INCLUDE=yes
# CPU_IDLE_ENABLED=yes
# FAST_BOOT_INCLUDE=yes
#
```

Binary size - AppImage_UcEarly_BE 4.5MB
AppImage_UcLate_BE 12.6 MB
Boot media – QSPI
Capture – OV10640P
Display – LCD (10 inch)
Algorithm – Object Detect

14.4 Boot time

14.4.1 POR to Display time split

No.	Description	Time
1.	SBL	167 ms
2.	Sensor initialization time with I2C 400 KHz	244 ms
3.	Time take by Framework	151 ms
	Power On Reset to Display Time	**562 ms**

As far as boot time is considered POR on to reset split is important which is mentioned above. If carefully observed measure time is spent in sensor initialization.

This can be further reduced by programming resister only needed, thus by reducing I2C read writes.

Overall vision_sdk takes 167 + 151 = **318 ms** only for capture + display usecase

14.4.2 POR to Object Detect

No.	Description	Time
1.	Power On to reset to Object Detect	**1320 ms**

This is the time when full Object Detect Usecase is functional since POR.

15 Stereo CameraUsecase on TDA2xx (MonsterCam)

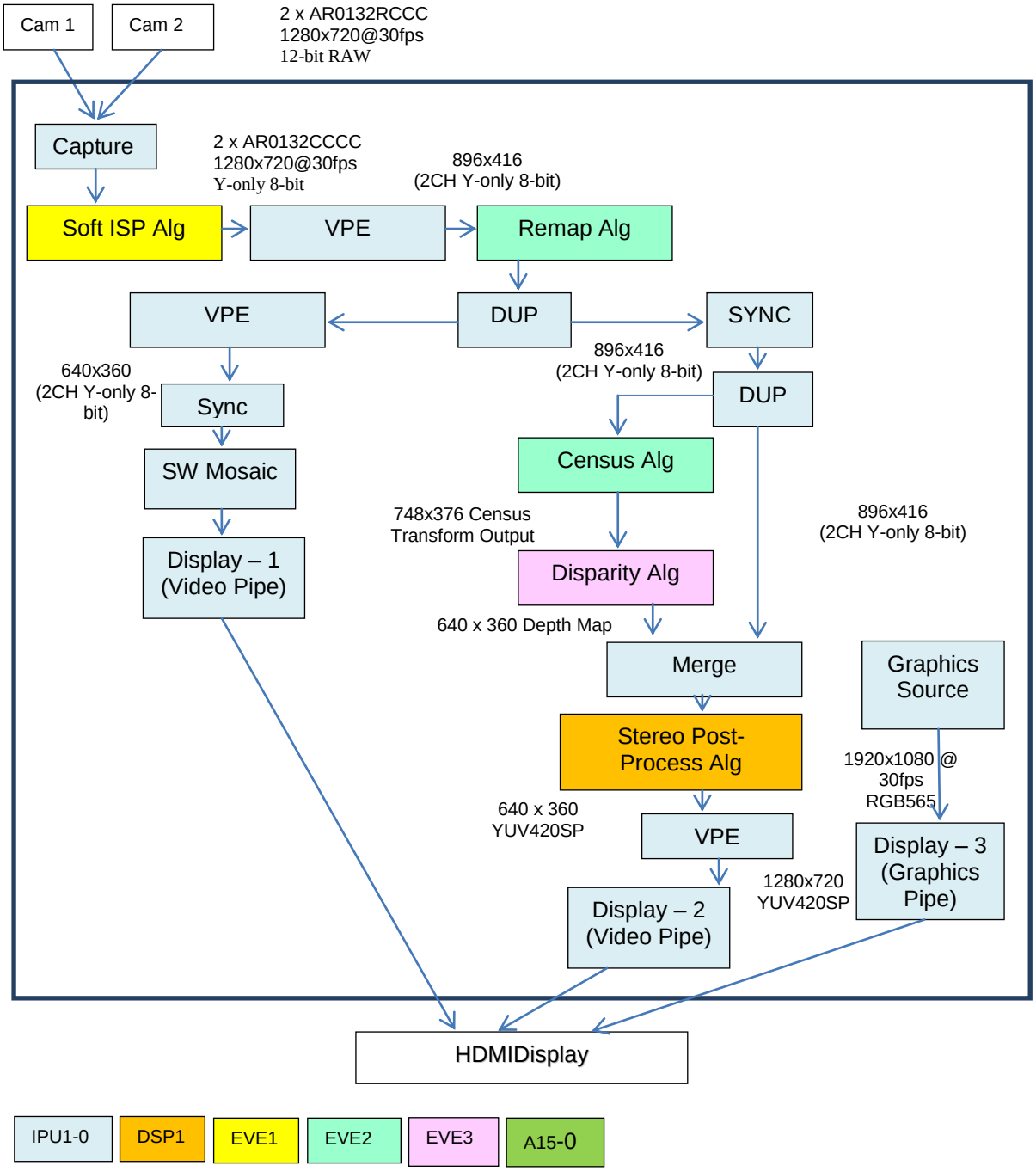
15.1 Overview

This configuration is used to demonstrate the system performance of running stereo demo on TDA2xx Monstercam platform

15.2 DataFlow

15.2.1 Stereo only usecase example

The Monstercam is equipped with 2 RCCC sensors used by the stereovision algorithm to produce a disparity map of the field of view of both sensors. Stereo data from two AR0132RCCC sensors is captured on 2 channels. The capture output is 720p, RCCC format. This 2 Channel data goes to Soft ISP linkwhich converts the 16 bit RCCC data to 8 bit 'C' frames. The output of Soft ISP is scaled down to 896x416 dimensions. This size is arrived at after taking into consideration the padded region required by all the downstream algorithms. The Remap Merge algorithm then uses rectification LUTs to correct lens distortions and misalignment between the data from left and right sensor. The output of Remap is Dup'ed and one of these is displayed as a 2x1 Mosaic. The other path is sent to Sync link which synchronizes the data on two channels. This is then Dupe'd and sent to Census link which applies a Census transform on both the channels. The output of Census link is 748x376 (2 channel data) goes to Disparity Haming Distance algorithm link which generates a 640x360 depth map. This depth map alongwith output of Remap algorithm is processed in Post-process link to remove artifacts. After scaling up to 720p using a VPE the depth map is then displayed on an HDMI display.



*IPC IN/OUT blocks are left-out to improve readability.

15.3 System Parameters

Refer to section 18.1 for common system parameters.

The benchmarks in this section are computed for HDMI display (1080p60) scenario.

15.4 CPU Loading and Task Info

15.4.1 Total CPU Load

CPU	LOAD TYPE	CPU LOAD (%)
IPU1_0	**HWI**	**4.9**
	SWI	**1.2**
	Total	**23.4**
DSP1	**HWI**	**0.6**
	SWI	**0.1**
	Total	**85.0**
EVE1	**HWI**	**2.2**
	SWI	**0.6**
	Total	**27.9**
EVE2	**HWI**	**1.9**
	SWI	**0.2**
	Total	**37.4**
EVE3	**HWI**	**0.2**
	SWI	**0.1**
	Total	**100**

15.4.2 Task Level Information and Task Level CPU Load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)
IPU1-0	Stat Collector	Statistics collector	2.2
	Capture	Capture frames from sensor via VIP port	0.4
	DUP (2 links)	Duplicates frame	0.4
	VPE (3 links)	Scaling frames	4.4
	Sync (2 links)	Sync frames based on timestamp from multiple channels	1.1
	Merge	Merge frames	0.1
	Display (3 links)	Display frames via DSS	1.3
	ALG – SW Mosaic	SW Mosaic ALgo	0.5
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	8.3

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)
	IPC OUT (4 links)	To send frame to another processor	2.8
	IPC IN (4 links)	To receive frames from another processor	0.9
DSP1	IPC + Stereo Post Process	IPC + Stereo Post process algorithm	84.1
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.7
EVE1	IPC + ALGSoft ISP	IPC + Soft ISP algorithm link	24.8
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.3
EVE2	IPC + ALG Remap + ALG Census	Remap Merge algorithm + Census ALg plugin + IPC	37.5
	UNKNOWN	This is unaccounted CPU load after subtracting the individual task load from total CPU Load	0.3
EVE3	IPC + ALG SOF	Sparse Optical Flow algorithm	100.0%

NOTE: CPU load of 0.0% means the CPU load was so low it could not be measured with sufficient granularity in load calculations

NOTE: There could be minor variations of +/-0.1% CPU load in different runs of the same use-case

NOTE: Graphics Load is not accounted for in these measurements.

NOTE: Other than above tasks few additional tasks as listed below are active in each processor

- Message Q task: This task is used to listen to control messages sent by any other core. Normally at run-time very few control messages are sent hence this task does not appear in the task load print log
- Processor link task: This task is used to send generic non-link specific messages to another CPU. Normally at run-time very few messages are sent to this task hence this task does not appear in the task load print log
- Remote log client task (on IPU1-0 only): This task looks at the shared remote log buffer and outputs any strings to UART terminal. Normally during run-time prints are not done hence this task does not appear in the task load log

15.5 System Performance

COMPONENT	PARAMETER	TDA2XX
Capture	Output fps	30fps on each of 2 channels
ALG – Soft ISP (EVE1)	Output fps	30fps
	Avg time per frame	3.7ms
ALG – Remap Merge (EVE2)	Output fps	30fps
	Avg time per frame	3.4ms
ALG – Census (EVE2)	Output fps	26fps
	Avg time per frame	5.4ms
ALG – Disparity (EVE3)	Output fps	26fps
	Avg time per frame	40ms
ALG – Stereo Post Process (DSP1)	Output fps	26fps
	Avg time per frame	35ms
Display VID 1	Input fps	25fps
	VENC fps	60fps

NOTE: FPS numbers are rounded off to nearest integer

15.6 System Memory Usage

15.6.1 Code/Data Memory Usage

Refer section 18.2for common Code/Data Memory usage

15.6.2 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	TDA2XX MEMORY SIZE USED
IPU1-0	Local Heap	256 KB	18 KB
	HDVPSS Descriptor Mem	1MB	705KB
DSP1	L2	223 KB	19 KB
	Local Heap	512 KB	17KB
EVE1	L2	24 KB	4 KB
	Local Heap	256KB	12 KB
EVE2	L2	24 KB	3 KB
	Local Heap	256KB	16 KB
EVE3	L2	24 KB	5 KB
	Local Heap	256KB	11 KB
Shared Memory	SR0	128KB	128KB
	Frame Buffer (SR1)	256MB	256MB
	SR2 OCMC	512 KB	0 KB
	Remote Log Buffer	160KB	158KB

15.6.3 DDR Bandwidth

PARAMETER	BANDWIDTH	TDA2XX
EMIF Read Only	Avg	1311 MB/s
	Peak	1871 MB/s
EMIF Write Only	Avg	452 MB/s
	Peak	1085 MB/s
EMIF Read + Write	Avg	1762 MB/s
	Peak	2489 MB/s

15.7 Other Benchmarks

15.7.1 Processing Latency

		TDA2XX LATENCY
Capture to Display Latency (Display VID 2)	Avg	225ms
	Min	108ms
	Max	263ms

NOTE:

- This latency is as measured inside the system by software.
- There will an additional 1/(capture rate) added on top of this from sensor/receiver itself.
- There will an additional 1/(display rate) added on top of this for the frame to actually get displayed on the screen.
- Thus e.g. in a scenario of display at 60fps and capture at 30fps - 16.67ms + 33.33ms needs to be added to latency figures in above table to get true capture to display latency
-

15.7.2 Boot Time

PARAMETER	TDA2XX DURATION
SBL to App main() (QSPI Boot) The boot time will change based on the mode of boot – SD, NOR flash or QSPI.	2.059 s
main() to Use-case create	0.335 s
Use-case create start to Live preview on display	0.466 s
Total Boot time	**2.86 s**

QSPI Boot time measurement done with TDA2xx ES1.1 samples.

GUI and Sensor initialization time not accounted for.

16 Inter processor communication (IPC) latency

This section lists the latency measured when doing inter processor communication in Vision SDK

16.1 System Parameters (TDA2xx)

Refer to section 17.1 for common system parameters. The benchmarks in this section are computed for OV10635 capture (720p30) and LCD display scenario.

Other important system parameters are listed below

COMPONENT	PROPERTY	VALUE
IPC	Interrupt mechanism	Notify with SHM transport
	Information exchange mechanism	Non-Cache shared region in DDR (SR0)
Measurement mechanism	Timer used	CLK32KHZ
	Measurement unit	usecs
	Duration of Measurement	30 secs
	Number of frame exchanges over which results are averaged	30*30 = 900

16.2 IPC Latency measurements

In order to measure latency from all CPUs to all CPUs, three use-cases were run as shown below. The color in measurement table shows the use-case in which the IPC latency was measured.

Usecase 1	Capture(IPU1_0) ->IPU1_0->DSP1->EVE1->EVE2->DSP2->EVE3->A15_0->EVE4->IPU1_1->IPU1_0 -> Display(IPU1_0)
Usecase 2	Capture(IPU1_0) ->IPU1_0->A15_0->DSP1->DSP2->IPU1_1->EVE1->IPU1_0 -> Display(IPU1_0)
Usecase 3	Capture(IPU1_0) ->IPU1_0->DSP1->A15_0->IPU1_0 -> Display(IPU1_0)

16.2.1 IPC Buffer Passing Latency measurement (TDA2xx)

This measures the time taken from the point a new buffer is received by IPC OUT link on SRC CPU to the point in IPC IN link on DSP CPU where the buffer is given to the next link in the processing chain.

Roughly this corresponds to the IPC Notify latency + the time taken to copy / translate buffer information from one CPU to another CPU via shared memory

The IPC buffer passing latency is shown below (All numbers in units of **micro-secs**)

DST \ SRC	A15	DSP	IPU	EVE
A15	NA	34	74	126
DSP	33	35	77	142
IPU	98	145	156	232
EVE	223	227	264	343

16.2.2 IPC Notify Latency Measurement (TDA2xx)

This measures the time taken from the point just before Notify_sendEvent() API is called on SRC CPU (in IPN OUT link) to the point (in IPC IN link) when SYSTEM_CMD_NEW_DATA command is received on DST CPU (NOTE: Notify callback (ISR) on DST CPU sends SYSTEM_CMD_NEW_DATA to IPC IN link on DST CPU).

Roughly this corresponds to IPC Notify send + Notify ISR + Task switch overhead.

The IPC Notify latency is shown below (All numbers in units of **micro-secs**)

DST \ SRC	A15	DSP	IPU	EVE
A15	NA	23	45	54
DSP	20	21	45	57
IPU	66	109	135	134
EVE	152	159	179	201

17 TDA2xx Common System Parameters

17.1 System Parameters

The system parameters mentioned below are common across all configurations unless specified otherwise.

COMPONENT	PROPERTY	VALUE
SOC	SOC Name	TDA2xx
	SOC revision	ES1.1 or ES1.0
EVM	EVM Name	TI TDA2xx EVM Vision Daughter card
IPU	Clock	212.8Mhz
	L1-P cache	ENABLED
	L1-D cache	ENABLED
	Code/Data Placement	DDR
DSP	Clock	600Mhz
	L1-P cache	32KB ENABLED
	L1 D cache	32KB ENABLED
	L2 P/D cache	32KB ENABLED
	Code/Data Placement	DDR
EVE	Clock	ARP32 267.5MHz VCOP 535Mhz
	L1-P cache	ENABLED
	DMEM	USED FOR IPC and ALG TASK STACK
	Code/Data Placement	DDR
A15-0	Clock	750Mhz
	P/D cache	ENABLED
	Code/Data Placement	DDR
DDR Config	Clock	532Mhz
	Bus Width	32-bit
	Number of EMIFs	1
	DDR size	1 GB
Sensor	Part number	OV10635

COMPONENT	PROPERTY	VALUE
	PCLK	74.25Mhz
	Resolution @ frame-rate	1280x720@30fps
	Data format	YUV422 interleaved
	Bus width	8-bit
	Sync Type	HS/VS discrete sync
HDMI Receiver 1 *	Part number	SII 9127
	Resolution @ frame-rate	1920x1080@60fps
	Data format	YUV422 interleaved
	Bus width	16-bit
	Sync Type	Discrete sync with AVID and VBLK control signals
HDMI Receiver 2 *	Part number	ADV 7611
	Resolution @ frame-rate	1920x1080@60fps
	Data format	YUV422 interleaved
	Bus width	16-bit
	Sync Type	Discrete sync with AVID and VBLK control signals
LCD 1 *	Part number	7-inch, WVGA LCD #703663
	DCLK	29.232Mhz
	Resolution @ frame-rate	800x480 @ 60fps
	Data format	RGB888
	Bus width	24-bit
	Sync Type	HS/VS discrete sync
LCD 2 *	Part number	10-inch, WXGA LCD #LG LP101WX2
	DCLK	74.5Mhz
	Resolution @ frame-rate	1280x800 @ 60fps
	Data format	RGB888
	Bus width	24-bit
	Sync Type	HS/VS discrete sync
DSS Display	DSS pipe	VID1 VID2

COMPONENT	PROPERTY	VALUE
		VID3 GRPX Any or all of above used based on use-case
	DSS output port	DPI1 for LCD and HDMI for HDMI display type
	DSS VENC	LCD1 or LCD2 for LCD and HDMI for HDMI display type
	Inline scaling	ENABLED or DISABLED based on use-case

* NOTE Above table lists all HDMI receivers and LCDs supported. But at a time only one type of LCD can be connected. Same applies to other devices.

17.2 Code/Data Memory Usage

NOTE: Code/data memory for data structures is same for all configurations and all use-cases since a single binary is used for all configurations and all use-cases. These configurations are with respect to 1GB Memory map.

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Initialized section (.text, .const)	6MB	5.1MB
	Uninitialized section (.bss, .heap, .stack)	14MB	8.14MB
IPU1-1	Initialized section (.text, .const)	2MB	227 KB
	Uninitialized section (.bss, .heap, .stack)	10MB	2.75MB
DSP1	Initialized section (.text, .const)	2MB	670 KB
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	64MB	8.17 MB
DSP2	Initialized section (.text, .const)	2MB	670 KB
	Uninitialized section (.bss, .heap, .stack)	64MB	8.67 MB
EVE1	Initialized section (.text, .const)	2MB	538 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
EVE2	Initialized section (.text, .const)	2MB	487 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	5.67 MB
EVE3	Initialized section (.text, .const)	2MB	449.8 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
EVE4	Initialized section (.text, .const)	2MB	449.8 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
A15-0	Initialized section (.text, .const) Uninitialized section (.bss, .heap, .stack)	16 MB	11MB

17.3 App Image Size

PARAMETER	VALUE
App Image size (9 CPU images)	29.3 MB

This App Image contains images for all the 9 processors.

18 TDA2xx-MC Common System Parameters

18.1 System Parameters

The system parameters mentioned below are common across all configurations unless specified otherwise.

COMPONENT	PROPERTY	VALUE
SOC	SOC Name	TDA2xx
	SOC revision	ES1.1
EVM	EVM Name	MonsterCam
IPU	Clock	212.8Mhz
	L1-P cache	ENABLED
	L1-D cache	ENABLED
	Code/Data Placement	DDR
DSP	Clock	600Mhz
	L1-P cache	32KB ENABLED
	L1 D cache	32KB ENABLED
	L2 P/D cache	32KB ENABLED
	Code/Data Placement	DDR
EVE	Clock	ARP32 267.5MHz VCOP 535Mhz
	L1-P cache	ENABLED
	DMEM	USED FOR IPC and ALG TASK STACK
	Code/Data Placement	DDR
A15-0	Clock	750Mhz
	P/D cache	ENABLED
	Code/Data Placement	DDR
DDR Config	Clock	532Mhz
	Bus Width	32-bit
	Number of EMIFs	1
	DDR size	1 GB
Sensor	Part number	AR0132AT6C00XPEA0
	PCLK	74.25MHz

COMPONENT	PROPERTY	VALUE
	Resolution @ frame-rate	720p60
	Data format	Bayer
	Bus width	12 bits
	Sync Type	DSC
Sensor	Part number	AR0132AT6R00XPEA0
	PCLK	38.8 MHz
	Resolution @ frame-rate	720p30
	Data format	RCCC
	Bus width	12 bit
	Sync Type	DSC
DSS Display	DSS pipe	VID1 VID2 VID3 GRPX Any or all of above used based on use-case
	DSS output port	HDMI for HDMI display type
	DSS VENC	HDMI for HDMI display type
	Inline scaling	ENABLED or DISABLED based on use-case

18.2 Code/Data Memory Usage

NOTE: Code/data memory for data structures is same for all configurations and all use-cases since a single binary is used for all configurations and all use-cases. These configurations are with respect to 1GB Memory map.

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Initialized section (.text, .const)	6MB	5.16 MB
	Uninitialized section (.bss, .heap, .stack)	14MB	8.14MB
IPU1-1	Initialized section (.text, .const)	2MB	227 KB
	Uninitialized section (.bss, .heap, .stack)	10MB	2.75 MB
DSP1	Initialized section (.text, .const)	2MB	669.5 KB
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	64MB	8.20 MB
DSP2	Initialized section (.text, .const)	2MB	669.5KB
	Uninitialized section (.bss, .heap, .stack)	64MB	8.67 MB
EVE1	Initialized section (.text, .const)	2MB	449.8 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
EVE2	Initialized section (.text, .const)	2MB	487 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	5.7 MB
EVE3	Initialized section (.text, .const)	2MB	449.8 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
EVE4	Initialized section (.text, .const)	2MB	449.8 KB
	Uninitialized section (.bss, .heap, .stack)	14MB	1.43 MB
A15-0	Initialized section (.text, .const) Uninitialized section (.bss, .heap, .stack)	16MB	11 MB

18.3 App Image Size

PARAMETER	VALUE
App Image size (9 CPU images)	29.3MB

This App Image contains images for all the 9 processors.

19 TDA2Ex- Common System Parameters

19.1 System Parameters

The system parameters mentioned below are common across all configurations unless specified otherwise.

COMPONENT	PROPERTY	VALUE
SOC	SOC Name	TDA2Ex
	SOC revision	ES1.0
EVM	EVM Name	TI TDA2Ex EVM Vision Daughter card
IPU	Clock	212.8MHz
	L1-P cache	ENABLED
	L1-D cache	ENABLED
	Code/Data Placement	DDR
DSP	Clock	600MHz
	L1-P cache	32KB ENABLED
	L1 D cache	32KB ENABLED
	L2 P/D cache	32KB ENABLED
	Code/Data Placement	DDR
A15-0	Clock	800MHz
	P/D cache	ENABLED
	Code/Data Placement	DDR
DDR Config	Clock	666Mhz
	Bus Width	32-bit
	Number of EMIFs	1
	DDR size	1 GB
Sensor	Part number	OV10635
	PCLK	74.25Mhz
	Resolution @ frame-rate	1280x720@30fps
	Data format	YUV422 interleaved
	Bus width	8-bit

COMPONENT	PROPERTY	VALUE
	Sync Type	HS/VS discrete sync
HDMI Receiver 1 *	Part number	SII 9127
	Resolution @ frame-rate	1920x1080@60fps
	Data format	YUV422 interleaved
	Bus width	16-bit
	Sync Type	Discrete sync with AVID and VBLK control signals
HDMI Receiver 2 *	Part number	ADV 7611
	Resolution @ frame-rate	1920x1080@60fps
	Data format	YUV422 interleaved
	Bus width	16-bit
	Sync Type	Discrete sync with AVID and VBLK control signals
LCD 1 *	Part number	10-inch, WXGA LCD #LG LP101WX2
	DCLK	74.5Mhz
	Resolution @ frame-rate	1280x800 @ 60fps
	Data format	RGB888
	Bus width	24-bit
	Sync Type	HS/VS discrete sync
DSS Display	DSS pipe	VID1 VID2 VID3 GRPX Any or all of above used based on use-case
	DSS output port	DPI1 for LCD and HDMI for HDMI display type
	DSS VENC	LCD1 or LCD2 for LCD and HDMI for HDMI display type
	Inline scaling	ENABLED or DISABLED based on use-case

* NOTE Above table lists all HDMI receivers and LCDs supported. But at a time only one type of LCD can be connected. Same applies to other devices.

19.2 Code/Data Memory Usage

NOTE: Code/data memory for data structures is same for all configurations and all use-cases since a single binary is used for all configurations and all use-cases. These configurations are with respect to 1GB Memory map.

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Initialized section (.text, .const)	6MB	5.16MB
	Uninitialized section (.bss, .heap, .stack)	14MB	6.36MB
IPU1-1	Initialized section (.text, .const)	2MB	226 KB
	Uninitialized section (.bss, .heap, .stack)	10MB	2.76 MB
DSP1	Initialized section (.text, .const)	2MB	669KB
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	64MB	8.17 MB
A15-0	Initialized section (.text, .const) Uninitialized section (.bss, .heap, .stack)	16MB	11 MB

19.3 App Image Size

PARAMETER	VALUE
App Image size (9 CPU images)	17.7 MB

This App Image contains images for all the 4 processors.

20 TDA3xx Common System Parameters

20.1 System Parameters

The system parameters mentioned below are common across all configurations unless specified otherwise.

COMPONENT	PROPERTY	VALUE
SOC	SOC Name	TDA3xx
	SOC revision	1.0
EVM	EVM Name	TI TDA3xx EVM
IPU	Clock	212.8Mhz
	L1-P cache	ENABLED
	L1-D cache	ENABLED
	Code/Data Placement	DDR
DSP	Clock	500Mhz
	L1-P cache	32KB ENABLED
	L1 D cache	32KB ENABLED
	L2 P/D cache	32KB ENABLED
	Code/Data Placement	DDR
EVE	Clock	ARP32 250MHz VCOP 500Mhz
	L1-P cache	ENABLED
	DMEM	USED FOR IPC and ALG TASK STACK
	Code/Data Placement	DDR
DDR Config	Clock	532Mhz
	Bus Width	32-bit
	Number of EMIFs	1
	DDR size	512 MB
Sensor 1 *	Part number	OV10635
	PCLK	74.25Mhz
	Resolution @ frame-rate	1280x720@30fps
	Data format	YUV422 interleaved
	Bus width	8-bit

COMPONENT	PROPERTY	VALUE
	Sync Type	HS/VS discrete sync
Sensor 2 *	Part number	OV10640 CSI2
	PCLK	90 MHz
	Resolution @ frame-rate	1280x720@23fps
	Data format	RAW Bayer
	Bus width	4 lanes
	Sync Type	HS/VS discrete sync
Sensor 3 *	Part number	OV10640 Parallel
	PCLK	90 MHz
	Resolution @ frame-rate	1280x720@23fps
	Data format	RAW Bayer
	Bus width	12-bit
	Sync Type	HS/VS discrete sync
Sensor 4 *	Part number	AR0132
	PCLK	74.25 MHz
	Resolution @ frame-rate	1280x720@60fps
	Data format	RAW Bayer
	Bus width	12-bit
	Sync Type	HS/VS discrete sync
Sensor 5 *	Part number	AR0140
	PCLK	74MHz
	Resolution @ frame-rate	1280x800 @ 30fps
	Data format	Bayer
	Bus width	12bit
	Sync Type	Discrete sync
HDMI Receiver *	Part number	ADV 7611
	Resolution @ frame-rate	1920x1080@60fps
	Data format	YUV422 interleaved
	Bus width	16-bit
	Sync Type	Discrete sync with AVID and VBLK control signals

COMPONENT	PROPERTY	VALUE
LCD 1 *	Part number	10-inch, WXGA LCD #LG LP101WX2
	DCLK	74.5Mhz
	Resolution @ frame-rate	1280x800 @ 60fps
	Data format	RGB888
	Bus width	24-bit
	Sync Type	HS/VS discrete sync
HDMI TX 1 *	Part number	SII 9022A
	DCLK	148.5Mhz
	Resolution @ frame-rate	1920x1080 @ 60fps
	Data format	RGB888
	Bus width	24-bit
	Sync Type	HS/VS discrete sync
SD Display 1	Part number	On-Chip SDDAC
	DCLK	27Mhz
	Resolution @ frame-rate	NTSC: 720x240 @ 60 fps (Interlaced) PAL: 720x288 @ 50 fps (Interlaced)
DSS Display	DSS pipe	VID1 VID2 GRPX Any or all of above used based on use-case
	DSS output port	DPI1 for LCD, HDMI (Off-Chip HDMI TX)
	DSS VENC	LCD1 for LCD and HDMI (Off-Chip HDMI TX) SDDAC for NTSC/PAL Display
	Inline scaling	ENABLED or DISABLED based on use-case

* NOTE Above table lists all Sensors, HDMI receivers and LCDs supported. But at a time only one type of sensor can be connected. Same applies to other devices.

20.2 Code/Data Memory Usage

NOTE: Code/data memory for data structures is same for all configurations and all use-cases since a single binary is used for all configurations and all use-cases. These configurations are with respect to 1GB Memory map.

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Initialized section (.text, .const)	4.5 MB	3.27 MB
	Uninitialized section (.bss, .heap, .stack)	14 MB	8.17 MB
IPU1-1	Initialized section (.text, .const)	2 MB	219 KB
	Uninitialized section (.bss, .heap, .stack)	12 MB	2.80 MB
DSP1	Initialized section (.text, .const)	2 MB	631KB
	Uninitialized section (.bss, .heap, .stack)	13 MB	7.25 MB
DSP2	Initialized section (.text, .const)	2 MB	631KB
	Uninitialized section (.bss, .heap, .stack)	13 MB	7.75 MB
EVE1	Initialized section (.text, .const)	2 MB	400 KB
	Uninitialized section (.bss, .heap, .stack)	14 MB	1.39 MB

20.3 App Image Size

PARAMETER	VALUE
App Image size (5 CPU images)	17.3 MB

This App Image contains images for all the 5 processors.

21 Revision History

Version	Date	Revision History
1.00	1 Oct 2013	Updated for Vision SDK release v2.01
2.00	10 Mar 2014	Updated for Vision SDK release v2.02
2.01	27 Mar 2014	Add section on IPC latency measurement
2.02	4th April 2014	Added Multi-channel AVB Surround view, Edge Detect Use case
2.03	31st July 2014	Updated for Vision SDK release v2.03
2.04	14th Nov 2014	Updated for Vision SDK release v2.05 for TDA2x and TDA3x usecases
2.06	4th March 2015	Updated for Vision SDK release v2.06 for TDA2x and TDA3x, TDA2Ex, TDA2x-MC usecases
2.07	7th July 2015	Updated for Vision SDK release v2.07 for TDA2x and TDA3x, TDA2Ex, TDA2x-MC usecases