VISION SDK LINUX (v03.07.00)

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

Note: Vertex and Fragment shaders are independent engines and it can run in parallel. VSDK system architecture allows this and hence the SGX load can be considered as the max of Fragment or Vertex Shader load. So, the GPU

1 Supported Features

1.1 Use-cases

The Linux + Vision SDK supports the following use-cases categories

Single Camera Use-cases

Multi-Camera LVDS Use-cases

OpenVx Use-case

AVB RX Use-case

ISS Use-case

From the above category, three use-cases are picked for system-parameter measurements in this document. For others data can be interpreted from BIOS use-cases results – please refer Vision SDK BIOS datasheet at $INSTALL_DIR\vision_sdk\docs\VisionSDK_DataSheet.pdf.

1.2 Framework features

- Refer $INSTALL_DIR\vision_sdk\docs\FeatureSpecificUserGuides\VisionSDK_Feature_List.xlsx

1.3 Supported Links on A15 + Linux

Links for the following modules with the features listed below

- Algorithm
- Dup, gate, merge, select, sync
- IPC_IN, IPC_OUT
- null, nullSrc
- sgxFrmcpy

2 Common System Parameters

Refer Section 2.1 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

When Linux is in picture all the cores run at OPP HIGH with the following clock frequencies:

CORE	TDA2xx Frequency MHz	TDA2Ex Frequency MHz	TDA2Px Frequency MHz
MPU	1000	1000	1000
DSPx	750	750	850
EVEx	650	NA	535
IPUx	212	212	212
GPU	532	532	665
IVA	532	532	532
DDR	532	666	666

2.1 TDA2xx - 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)	8MB	
	Uninitialized section (.bss, .heap, .stack)	21MB	
IPU2	Initialized section (.text, .const)	8MB	
	Uninitialized section (.bss, .heap, .stack)	20MB	
DSP1	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	24MB	
DSP2	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	14MB	
EVE1	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
EVE2	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
EVE3	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
EVE4	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
A15-0	Initialized section (.text, .const) Uninitialized section (.bss, .heap, .stack)	20MB	

2.2 TDA2Px - 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)	8MB	
	Uninitialized section (.bss, .heap, .stack)	21MB	
IPU2	Initialized section (.text, .const)	10MB	
	Uninitialized section (.bss, .heap, .stack)	20MB	
DSP1	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	24MB	
DSP2	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	10MB	
EVE1	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
EVE2	Initialized section (.text, .const)	2MB	
	Uninitialized section (.bss, .heap, .stack)	13.5MB	
A15-0	Initialized section (.text, .const)	20MB	

2.3 TDA2Ex - 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.

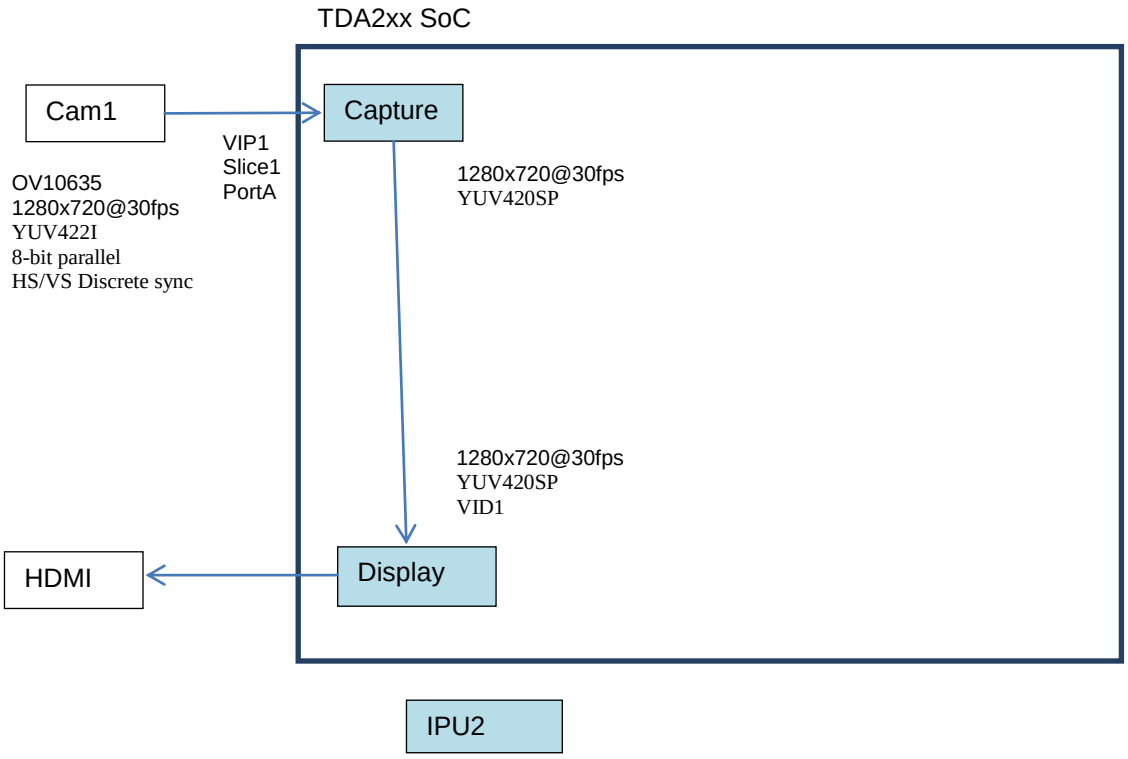
CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED
IPU1-0	Initialized section (.text, .const)	8MB	5.16MB
	Uninitialized section (.bss, .heap, .stack)	20MB	6.36MB
IPU2	Initialized section (.text, .const)	8MB	
	Uninitialized section (.bss, .heap, .stack)	20MB	
DSP1	Initialized section (.text, .const)	2MB	669KB
	Uninitialized section (.bss, .heap, .stack, .far, .fardata)	24MB	8.17 MB
A15-0	Initialized section (.text, .const) Uninitialized section (.bss, .heap, .stack)	20MB	11 MB

3 1CH VIP capture + SGX Copy + DISPLAY usecase on TDA2xx & TDA2Ex

3.1 Overview

This usecase consists of continuous capture on IPU2 and display on IPU2 with SGX. Capture can be done at 720p@30fps (OV Sensor) via the VIP1 Slice1 Port A. Display can be on HDMI display via HDMI output port. This is supported on both TDA2x and TDA2Ex.

3.2 Dataflow



3.3 System Parameters

Refer Section 2.1 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

The only difference when Linux is in picture is, A15 runs at 1000MHz instead of 750 MHz

3.4 CPU loading and Task Info

3.4.1 Total CPU load

CPU	CPU LOAD (%)	
SOC	**TDA2xx**	**TDA2EX**
IPU2	14.7%	9.8%
Sgx	Fragment Shader – 18.1% Vertex Shader -0.75%	Fragment Shader – 21% Vertex Shader – 0.72%

3.4.2 Task Level Information and Task Level CPU load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)	
SOC			**TDA2xx**	**TDA2EX**
IPU2	Capture	Capture frames via VIP port	0.2%	0.2%
	Display	Display via sgx link	0.6%	0.6%
	IPC OUT	To send frame to another processor	0.6%	0.7%
	Stat Coll	Stat collector	2.9%	2.2%
	GrpxSrc0	Graphic source link	6.8%	2.4%
	App Ctrl	Sensor init and board level control	0.1%	0.3%
	Misc	Miscellaneous	0.6%	0.5%

3.4.3 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED	
SOC		**TDA2xx**	**TDA2XX**	**TDA2EX**
IPU2	Local heap	256KB	14KB	8KB
	HDVPSS Descriptor Mem	1MB	1MB	1MB

Shared Memory	SR0 DDR	1MB	189KB	9KB
	SR1 (Frame Buffers)	250MB	44MB	44MB
	Remote Log Buffer	256KB	145KB	145 KB

3.5 System Performance

COMPONENT	PARAMETER	VALUE	
SOC		**TDA2xx**	**TDA2EX**
Capture	Out FPS	30	30
SGX Copy	Out FPS	30	30
Display	Input FPS	30	30

NOTE: FPS numbers are rounded off to nearest integer

3.6 Processing Latency

		LATENCY	
SOC		**TDA2xx**	**TDA2Ex**
Capture to Display Latency	Avg	0.33ms	0.33ms
	Min	0.30ms	0.30ms
	Max	0.42ms	0.61ms

Note: Latency is from VIP output to the display. Additional capture latency needs to be added for end to end latency

3.7 DDR BW usage

PARAMETER	BANDWIDTH	TDA2XX	TDA2EX
EMIF1 Read + Write	Avg	553 MB/s	1103 MB/s
	Peak	1315 MB/s	2562 MB/s
EMIF2 Read + Write	Avg	583 MB/s	NA
	Peak	1418 MB/s	NA

4 Multi-channel 1MP 3D Surround view using SGX on Linux on TDA2xx & TDA2Ex

4.1 Overview

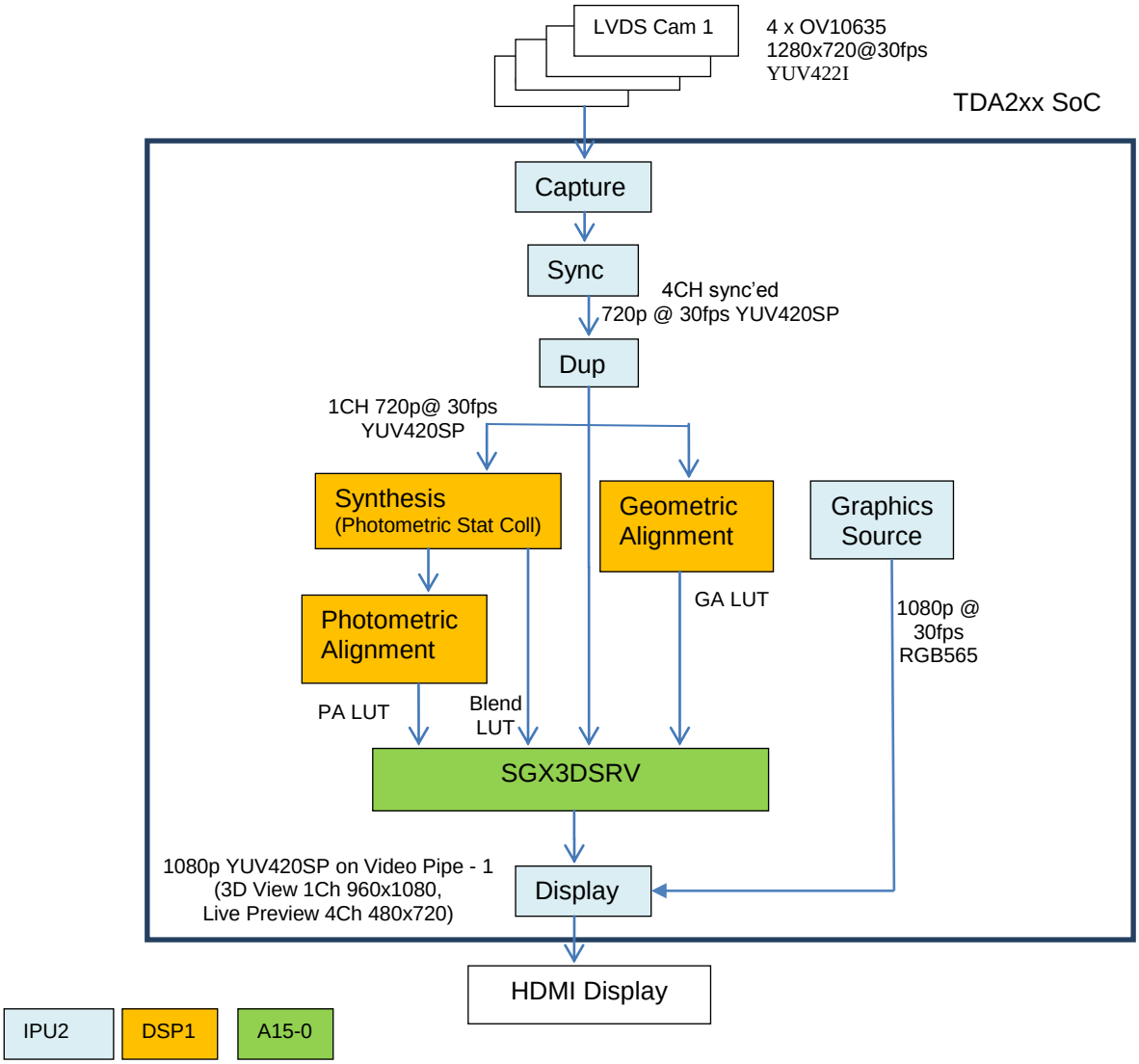
This use case demonstrates 1MP 3D surround view using SGX with Linux running on A15. This usecase is supported on both TDA2x and TDA2Ex.

4.2 Data Flow

4.2.1 4CH LVDS capture, 3D Surround View demonstration

In this configuration we capture 4 Channel video from 4 OV1063x sensors @ 720p 30fps. Captured frames are then passed on to Sync link, where these captured frames are synced based on time stamps. This "group of sync'ed frames" is passed on to algorithm links of Geometric alignment and synthesis link. Geometric Analysis link running on DSP core is invoked once in K frames. This link generates the geometric alignment LUTs to be used during synthesis stage. Photometric Analysis link runs on DSP core. Certain image statistics needed for calculating photometric LUTs are provided by the synthesis stage. Photometric Analysis link generates LUTs for pixel value transformations during Synthesis. Synthesis link on DSP encapsulates the algorithm for generating the photometric statistics & Blend Lookup tables, based on geometric and photometric LUTs. SGX link encapsulates the algorithm for stitching, based on geometric, Blend and photometric LUTs. This Link runs on A15 (Linux) and SGX (3D GRPX Engine) is used to create the 360 degree of the car using Open GL API/Algos. Output of Synthesis link is the stitched frame which is passed onto display link, which shall display the surround view image using display drivers via HDMI output.

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.

4.3 System Parameters

Refer Section 2.1 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

The only difference when Linux is in picture is, A15 runs at 1000MHz instead of 750 MHz

4.4 CPU loading and Task Info

4.4.1 Total CPU load

CPU	TDA 2xx CPU LOAD (%)	TDA 2ex CPU LOAD (%)
IPU2	30.0%	28.0%
DSP1	4.2%	4.2%
Sgx	Fragment Shader – 25% Vertex Shader – 11%	Fragment Shader – 37% Vertex Shader – 16%

4.4.2 Task Level Information and Task Level CPU load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)	
SOC			TDA2xx	TDA2ex
IPU2	Capture	Capture frames via VIP port	1.1%	1.1%
	Display	Display via sgx link	1.8%	1.7%
	VPE (2 Links)		4.7%	4.4%
	IPC OUT	To send frame to another processor	2.2%	2.2%
	IPC IN		0.4%	0.4%
	DUP Link		1.3%	1.3%
	SYNC Link	Sync Link to synchronize the frames from different channels	2.9%	3.0%
	Select Link		0.6%	0.5%
	Stat Coll	Stat collector	3.7%	2.4%
	GrpxSrc0	Graphic source link	1.9%	1.1%
	App Ctrl	Sensor init and board level control	0.1%	0.1%
	Misc	Miscellaneous	1.2%	1.1%

4.4.3 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	TDA 2xx MEMORY SIZE USED	TDA 2ex MEMORY SIZE USED
IPU2	Local heap	256KB	20KB	14KB
	HDVPSS Descriptor Mem	1MB	1MB	1MB
DSP 1	L2	221KB	128KB	128KB
	Local Heap	512KB	10KB	4KB
Shared Memory	SR0	1MB	189KB	9KB
	SR1 (Frame Buffers)	250MB	240MB	240MB
	Remote Log Buffer	256KB	145KB	145KB

4.5 System Performance

COMPONENT	PARAMETER	VALUE
Capture	Out FPS	30fps
ALG Synthesis (DSP1)	Output fps	30fps
ALG – DMA_SWMS (IPU1_0)	Output fps	30fps
SGX3DSRV (A15)	Output fps	30fps
ALG_PHOTOALIGN	Output fps	30fps
DISPLAY	Input FPS	30fps

NOTE: FPS numbers are rounded off to nearest integer.

4.6 Processing Latency

		LATENCY	
SOC		TDA2xx	TDA2Ex
Capture to Display Latency	Avg	13.8ms	9.87ms
	Min	13.4ms	8.9ms
	Max	30.6ms	21.4ms

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 DDR BW usage

PARAMETER	BANDWIDTH	TDA2XX	TDA2EX
EMIF1 Read + Write	Avg	691 MB/s	1409 MB/s
	Peak	1150 MB/s	2386 MB/s
EMIF2 Read + Write	Avg	721 MB/s	NA
	Peak	1166 MB/s	NA

5 Multi-channel 2MP 3D Surround view using SGX on Linux on TDA2Px

5.1 Overview

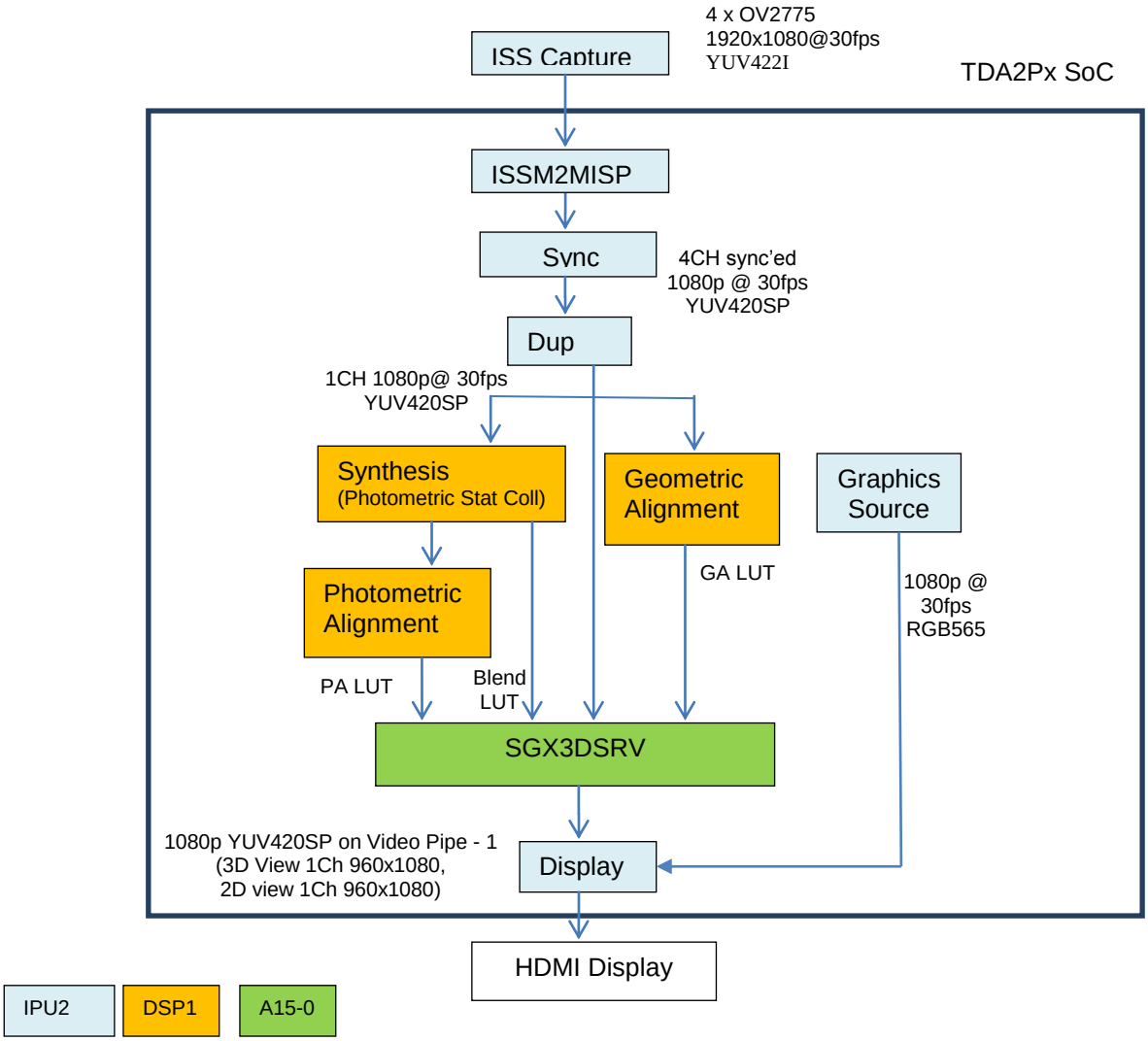
This use case demonstrates 2MP 3D surround view using SGX with Linux running on A15. This usecase is supported on TDA2Px

5.2 Data Flow

5.2.1 4CH LVDS capture, 3D Surround View demonstration

In this configuration we capture 4 Channel video from 4 OV2775 sensors @ 1080p 30fps. Captured frames are then passed on to Sync link, where these captured frames are synced based on time stamps. This "group of sync'ed frames" is passed on to algorithm links of Geometric alignment and synthesis link. Geometric Analysis link running on DSP core is invoked once in K frames. This link generates the geometric alignment LUTs to be used during synthesis stage. Photometric Analysis link runs on DSP core. Certain image statistics needed for calculating photometric LUTs are provided by the synthesis stage. Photometric Analysis link generates LUTs for pixel value transformations during Synthesis. Synthesis link on DSP encapsulates the algorithm for generating the photometric statistics & Blend Lookup tables, based on geometric and photometric LUTs. SGX link encapsulates the algorithm for stitching, based on geometric, Blend and photometric LUTs. This Link runs on A15 (Linux) and SGX (3D GRPX Engine) is used to create the 360 degree of the car using Open GL API/Algos. Output of Synthesis link is the stitched frame which is passed onto display link, which shall display the surround view image using display drivers via HDMI output.

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.

5.3 System Parameters

Refer Section 2.1 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

The only difference when Linux is in picture is, A15 runs at 1000MHz instead of 750 MHz

5.4 CPU loading and Task Info

5.4.1 Total CPU load

CPU	TDA 2Px CPU LOAD (%)
IPU2	28.8%
DSP1	6.2%
With SGX single core @665 MHz	Fragment Shader – **75%** Vertex Shader – 19%
With SGX dual cores @665 MHz	Fragment Shader – **50%** Vertex Shader – 14%

Note: Vertex and Fragment shaders are independent engines and it can run in parallel. VSDK system architecture allows this and hence the SGX load can be considered as the max of Fragment or Vertex Shader load. So, the GPU loading of 4xch 2MP split view surround view demo is 50% with dual core and 75% with single core.

5.4.2 Task Level Information and Task Level CPU load

CPU	TASK NAME	TASK DESCRIPTION	CPU LOAD (%)
SOC			**TDA2Px**
IPU2	ISS Capture	Capture frames via VIP port	0.9%
	Display	Display via sgx link	0.7%
	IPC OUT	To send frame to another processor	1.9%
	IPC IN		0.4%
	DUP Link		0.3%
	SYNC Link	Sync Link to synchronize the frames from different channels	1.1%
	Stat Coll	Stat collector	3.5%
	GrpxSrc0	Graphic source link	3.1%
	App Ctrl	Sensor init and board level control	0.1%
	Misc	Miscellaneous	0.6%

5.4.3 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	TDA 2Px MEMORY SIZE USED
IPU2	Local heap	256KB	105KB
	HDVPSS Descriptor Mem	1MB	1MB
DSP 1	L2	221KB	128KB
	Local Heap	512KB	7KB
Shared Memory	SR0	1MB	189KB
	SR1 (Frame Buffers)	330MB	285MB
	Remote Log Buffer	256KB	145KB

5.5 System Performance

COMPONENT	PARAMETER	VALUE
Capture	Out FPS	30fps
ALG Synthesis (DSP1)	Output fps	30fps
SGX3DSRV (A15)	Output fps	30fps
ALG_PHOTOALIGN	Output fps	30fps
DISPLAY	Input FPS	30fps

NOTE: FPS numbers are rounded off to nearest integer.

5.6 Processing Latency

		LATENCY
SOC		**TDA2Px**
Capture to Display Latency	Avg	0.37ms
	Min	0.30ms
	Max	2.34ms

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 DDR BW usage

PARAMETER	BANDWIDTH	TDA2PX
EMIF1 Read + Write	Avg	1514 MB/s
	Peak	2405 MB/s
EMIF2 Read + Write	Avg	1525 MB/s
	Peak	2425 MB/s

6 Multi-channel AVB capture and display usecase on Linux for TDA2xx & TDA2Ex

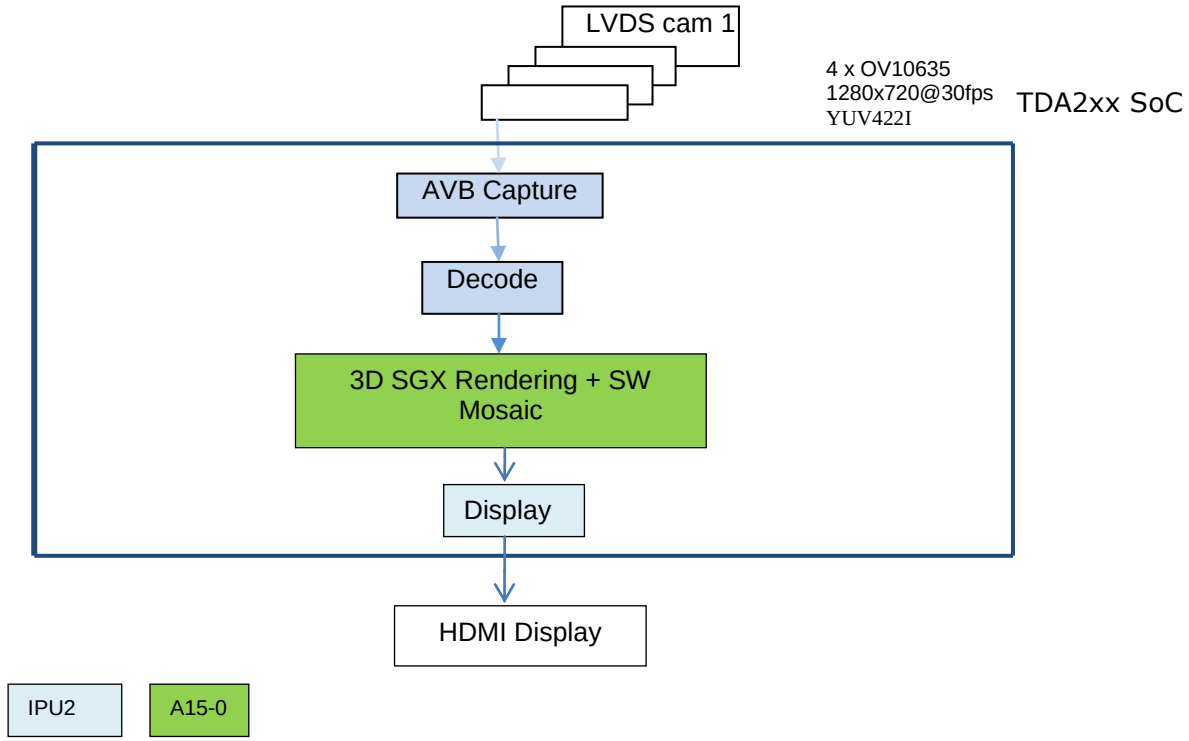
6.1 Overview

This use case demonstrates 4CH AVB using SGX with Linux running on A15. This is validated on TDA2x & TDA2Ex.

6.2 Data Flow

6.2.1 4CH AVB capture

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.

6.3 System Parameters

Refer Section 2.1 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

The only difference when Linux is in picture is, A15 runs at 1000MHz instead of 750 MHz.

6.4 CPU loading and Task Info

6.4.1 Total CPU load

CPU	TDA 2xx CPU LOAD (%)	TDA 2ex CPU LOAD (%)
IPU2	45.8%	42.9%
Sgx	Fragment Shader – 27.4% Vertex Shader – 0.85%	Fragment Shader – 37.1% Vertex Shader – 0.78%

6.4.2 Task Level Information and Task Level CPU load

CPU	TASK NAME	TASK DESCRIPTION	TDA 2xx CPU LOAD (%)	TDA 2ex CPU LOAD (%)
IPU2	IPC_OUT	To send frame to another processor	0.9%	0.8%
	IPC IN	To receive frame from another processor	0.5%	0.5%
	Display		0.9%	0.8%
	Decoder		4.2%	3.7%
	SYNC	Sync across frames from multiple channels	2.3%	2.1%
	STAT COLL	DDR BW stats collector	4.5%	3.3%
	MISC	Miscellaneous	12.7%	13.0%

6.4.3 Heap Memory Usage

CPU	MEMORY SECTION	MEMORY SIZE RESERVED	MEMORY SIZE USED on TDA2xx	MEMORY SIZE USED on TDA2ex
IPU2	Local heap	256KB	41KB	36KB
	HDVPSS Descriptor Mem	1MB	1MB	1MB
Shared	SR0	1MB	0MB	0MB

Memory	SR1 (Frame Buffers)	250MB	80MB	80MB
	Remote Log Buffer	256KB	145KB	145KB

6.5 System Performance

COMPONENT	PARAMETER	VALUE
Capture	Out FPS	31fps
DISPLAY	Input FPS	31fps

NOTE: FPS numbers are rounded off to nearest integer

6.6 Processing Latency

LATENCY		TDA2xx	TDA2ex
Capture to Display Latency	Avg	23.5ms	26.5ms
	Min	22.5ms	25.4ms
	Max	33.1ms	34.9ms

Note: Latency is from VIP output to the display. Additional capture latency needs to be added for end to end latency

6.7 DDR BW usage

PARAMETER	BANDWIDTH	TDA2XX	TDA2EX
EMIF1 Read + Write	Avg	750 MB/s	1435 MB/s
	Peak	1386 MB/s	2152 MB/s
EMIF2 Read + Write	Avg	721 MB/s	NA
	Peak	1310 MB/s	NA

7 System Memory Usage for Tda2xx

7.1 Code / Data Memory Usage

Overall Linux on A15 has full 1 GB on the EVM except following two holes in the memory

From 0x84203000 size 0x14A00000 bytes (vsdk_sr1_mem)
From 0xA0100000 size 0x100000 bytes (vsdk_sr0_mem)
From 0xA9000000 size 0x4000000 bytes (cmem)
From 0xA5000000 size 0x4000000 bytes (vsdk_eve_mem)

Refer Section 2.2 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf of code and data memory usage of IPU1_0

8 System Memory Usage for TDA2Ex

8.1 Code / Data Memory Usage

Overall Linux on A15 has full 1 GB on the EVM except following one holes in the memory

From 0x84203000 size 0x14A00000 bytes (vsdk_sr1_mem)
From 0xA0100000 size 0x100000 bytes (vsdk_sr0_mem)
From 0xA9000000 size 0x4000000 bytes (cmem)

Refer Section 4.2 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf of code and data memory usage of IPU1_0

9 IPC latency measurements

This section focuses on IPC latency measurements between Linux (A15) <-> Bios (IPU2). A15 <-> DSP IPC latencies are not measured.

For Bios <-> Bios ipc measurements refer Section 16.2 in $INSTALL_DIR/vision_sdk/docs/VisionSDK_DataSheet.pdf

9.1 A15->IPU2 IPC latencies

These are average case latencies.

This involves send from A15 + ISR on IPU2 and receipt of notification / message on IPU2

SOC	IPC One-way Notify Interrupt Latency (usecs)	IPC One-way Buffer Passing Latency (usecs)
TDA2xx	128	191
TDA2EX	129	194

9.2 IPU2->A15 IPC latencies

These are average case latencies.

This involves time for command msg sent using System_rpmsgSendNotify()/System_rpmsgMsgQSendMsg() on IPU2 till it is received on A15.

It means, it is send on IPU2 + ISR on A15 + scheduling / context switching overheads on Linux.

SOC	IPC One-way Buffer Passing Latency (usecs)	IPC One-way Notify Interrupt Latency (usecs)
TDA2xx	185	120
TDA2EX	233	169

10 Revision History

Version	Date	Revision History
1.0	25th July 2014	Created for Vision SDK release v2.03
2.0	17th Nov 2014	Updated for Vision SDK release v2.05
3.0	4th Mar 2015	Updated for Vision SDK release v2.06
4.0	7th July 2015	Updated for Vision SDK release v2.07
5.0	15th Oct 2015	Updated for Vision SDK release v2.08
6.0	29th Mar 2016	Updated for Vision SDK release v2.09
7.0	6th July 2016	Updated for Vision SDK release v2.10
8.0	31st Oct 2016	Updated for Vision SDK release v2.11
9.0	6th Feb 2017	Updated for Vision SDK release v2.12
10.0	14th Apr 2017	Updated for Vision SDK release v2.12.2 patch release Added 2MP SRV & Car Black Box loads/BW usage
11.0	04th July 2017	Updated for Vision SDK release v3.0
12.0	14th Oct 2017	Updated for Vision SDK release v3.1
13.0	21st Dec 2017	Updated for Vision SDK release v3.2
14.0	5th Apr 2018	Updated for Vision SDK release v3.3
15.0	2nd July 2018	Updated for Vision SDK release v3.4
16.0	7th June 2019	Updated for Vision SDK release v3.7