ORBiT AT SU®

OA-II VEH Camera System Design

DR00002

Rev: A04

Jinzhi Cai

2019–08–26

Contents

1 Introduction

1.1 Scope

This document discusses current camera technology and outlines a design that will fulfill the needs for OA-II VEH.

1.2 Purpose

The goal for this document is to come up with a design that will provide four 1080p 60Hz video steams and store that data to a central storage media by researching current options in camera technology.

1.3 Revision History

Rev#	Editor	Delta	Date
A01	Jinzhi Cai	Initialize	2019-7-13
A02	Jinzhi Cai	Add detail	2019-7-15
A03	Jinzhi Cai	Add Jetson Nano	2019-7-24
A04	Gabriel Smolnycki	Minor grammar edits	2019-08-26

Table 1: Summary of Revision History

2 General Structure of Camera System

2.1 Introduction

The most basic camera system has three parts. The camera sensor is the device that will receive data from the environment and transfer it via the camera interface. The data that flows out from the sensor is called raw data. It contains all the information that the camera takes in from the sensor. A 1080p 60Hz camera has $1920 * 1080 = 2073600 pixels$, and one pixel usually take $3Bytes$ to save. For each second, $2073600*3*60 = 373248000Bytes$ will be created. That will be $355.96MB/s$ for one single camera. An encoder is used to compress the video steam smaller for it to be transmitted via long distance data link(usually with in $1MB/s$).[1] The storage unit is then used to save the video steam to a file for future replay.

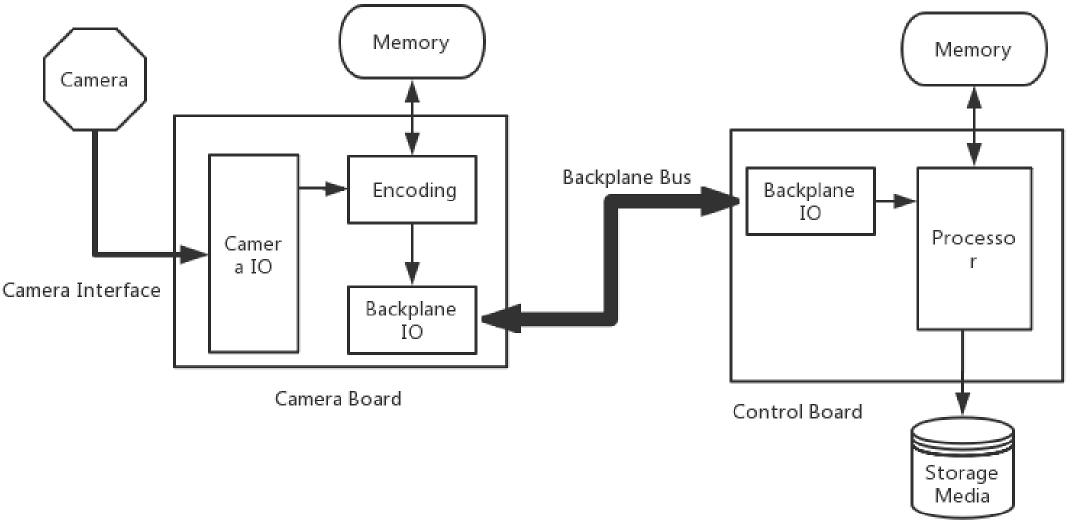

Figure 1: General Structure of Camera System

3 Camera Sensor and Relative Interface

3.1 Introduction

The sensor is the most important part of the system. Using a classical implementation, in the same way as a monitor would, data is output as video signal to an external device. For each pixel clock, data of a pixel has been transmitted. After few clock(depend on the resolution), Line Sync rise to indicate one line data is finish transmission. The Frame Sync is indicate a frame of pixel data is sent. However, modern camera sensor usually use one of those three camera interface. Some of the is similar with the classic one, and other improve it.

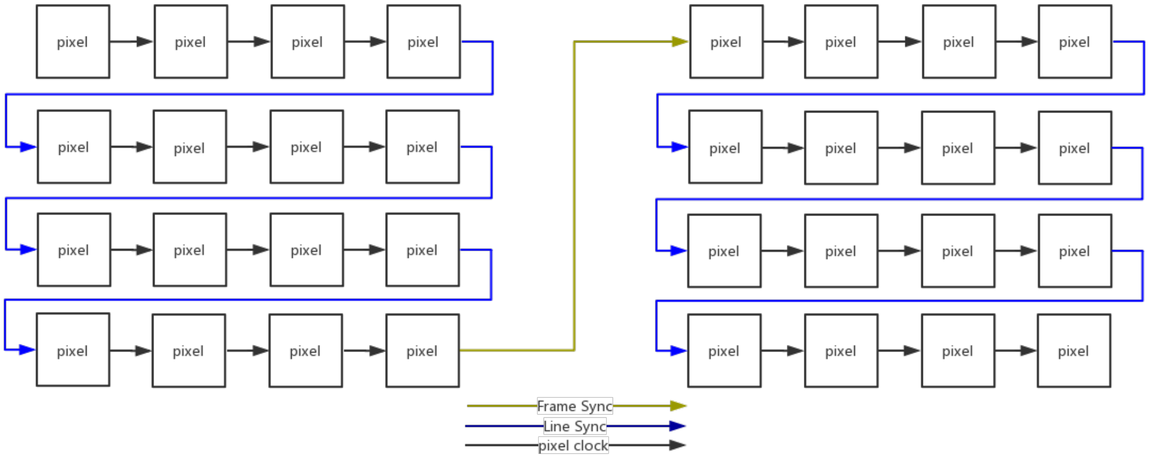

Figure 2: Classic Video Stream

3.2 DVP Interface

The DVP interface is use the classic video interface to transmit data out from the sensor. The I2C bus(SCK, SDA) that inside the interface is use to configure the camera. The main clock is use to provide a clock for the camera. The VSYNC, HSYNC, PCLK is the clock system that help processor to indicate the weight and height of frame. The D OUT is data that coming out from the camera. It have low requirement on the trace and sample interface design. However, it can not offer long distance transmission and high speed clock. It usually use at low end 240p camera.[2]

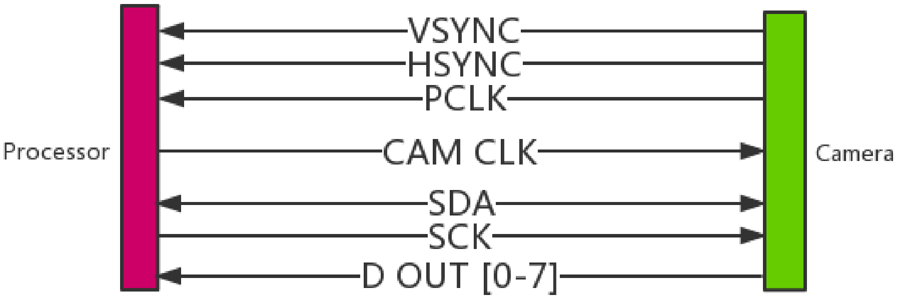

Figure 3: Physical Layer for DVP Interface

D OUT Data out put

VSYNC Lane Sync Signal

HSYNC Frame Sync Signal

PCLK Pixel Clock

CAM CLK Main Camera Clock

SDA I2C Data Line

SCK I2C Clock Line

3.3 LVDS Interface

The LVDS camera interface is use LVDS technology to improve classic DVP interface. It use differential signal to transmitting data. It increase the lane it need to send data, but differential signal can prevent common mode interference which limit the DVP clock for increasing.[3]

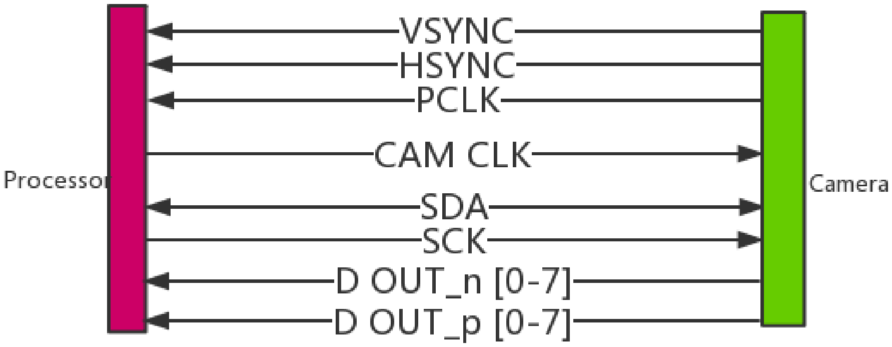

Figure 4: Physical Layer for LVDS Interface

DOUT n/p Differential Data Lane

VSYNC Lane Sync Signal

HSYNC Frame Sync Signal

PCLK Pixel Clock

CAM CLK Main Camera Clock

SDA I2C Data Line

SCK I2C Clock Line

3.4 MIPI CSI-2 Interface

The MIPI CSI-2 camera interface[1] is different from the old one. The goal for this bus is to cut down the lane count for camera module and allow more data flow to the processor. In structure, the MIPI CSI-2 camera interface use different way to communicate. The MIPI interface use package as the data communication unit. Inside the sensor, data will be pack into a package and send it via the two or four data lane. The same process will reverse and output the raw camera data. MIPI camera will allow 1080p 60Hz or 4K 30Hz.[4]

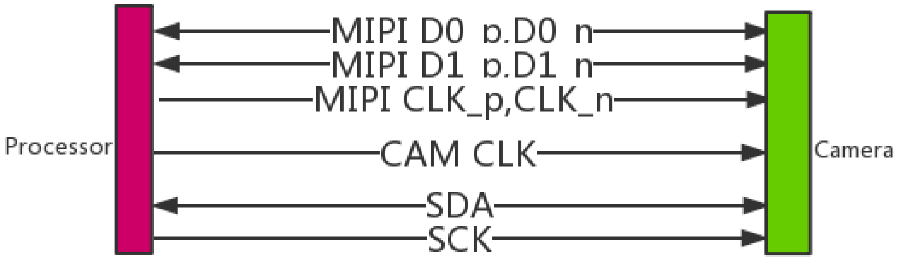

Figure 5: Physical Layer for 2 Lane MIPI Interface

MIPI D0n/p MIPI Data Lane 0

MIPI D1n/p MIPI Data Lane 1

MIPI CLKn/p MIPI Clock

CAM CLK Main Camera Clock

SDA I2C Data Line

SCK I2C Clock Line

3.5 USB Interface (UVC)

The USB camera interface[2] is different with the rest of the interface. In those camera, it include DSP chip that already finish the video compression.[5] It is easy to use and support 60fps.[3]

[1] Might only available in high-end chip.

[2] It do have success example by using ZYNQ SoC FPGA.

[3] With High End SoC chip

4 Camera Interface Bridge Chip

4.1 Lattice CrossLink

The CrossLink is Lattice made small scale MIPI bridge FPGA.[4] It include two MIPI D-PHY and up to 8 MIPI data lane. It also support interface translation (MIPI CSI-2 -> DVP).[6]

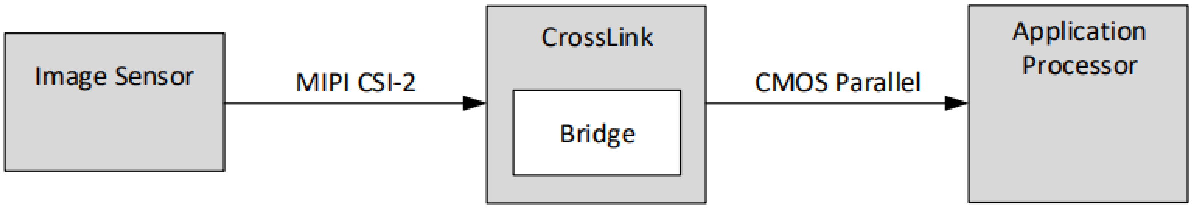

Figure 6: Block Diagram for CrossLink

4.2 Toshiba Camera Interface Bridge

The Toshiba Camera Interface Bridge is a stand alone chip that can converse MIPI CSI-2 bus to DVP bus. It also offer I2C bus to configure chip setting.[7]

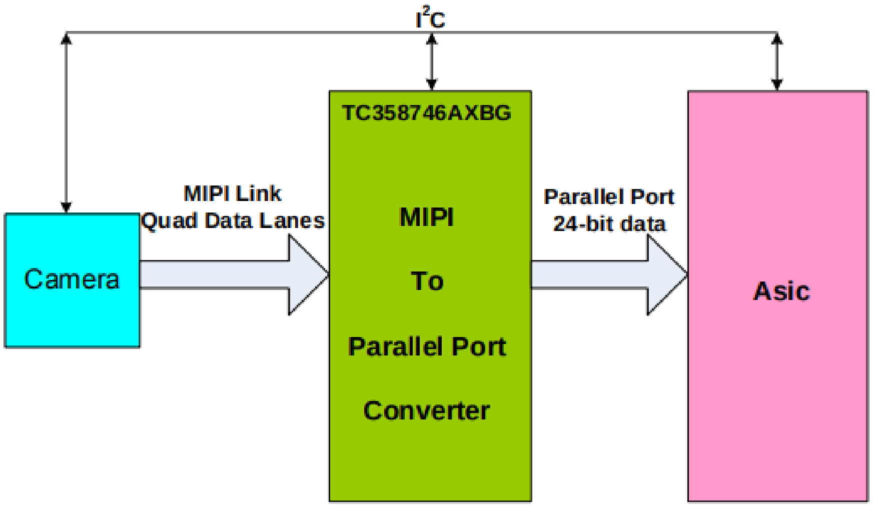

Figure 7: Block Diagram for Toshiba

[4] Too small for hardware encoder

5 H.264 Video Steam Encoding

5.1 Introduction

The H.264 Encoder is another important unit in the camera system. Due to the data flow of a camera sensor is too much for the backplane bus, compression is necessary.

5.2 IP Core Description

5.2.1 SoC Technology

This IP core fit for both **Intel** and **Xilinx** [8] [9].

Version	LUT use	Test Platform	Price for Platform
Standard Version	110k	Zynq-7 Z7045	$1000
Slim Version	50k	Artix-7 XC7A200T	$300
I-Frame Version	45k	Spartan-6 LX150	$200

Table 2: Encoder Summary

5.2.2 A2E Technology

Version	LUT use	Test Platform	Price for Platform
Xilinx(1080p 30Hz)	11K	Zynq-7 Z7020	$300
Xilinx(1080p 60Hz)	11K	Kintex-7	$500
Xilinx(1080p 180Hz)	11K	Zynq UltraScale	$1000
Intel(1080p 30Hz)	8K	Cyclone V SoC	$500

Table 3: SoC Summary

5.3 Encoding SoC Chip

5.3.1 HiSilicon Hi3559

Hi3559 is HiSilicon made new generation mobile camera soc. It support MIPI interface and H.264 Video encoding up to 1080P 60Hz. It can output video via USB 2.0 bus.[10]

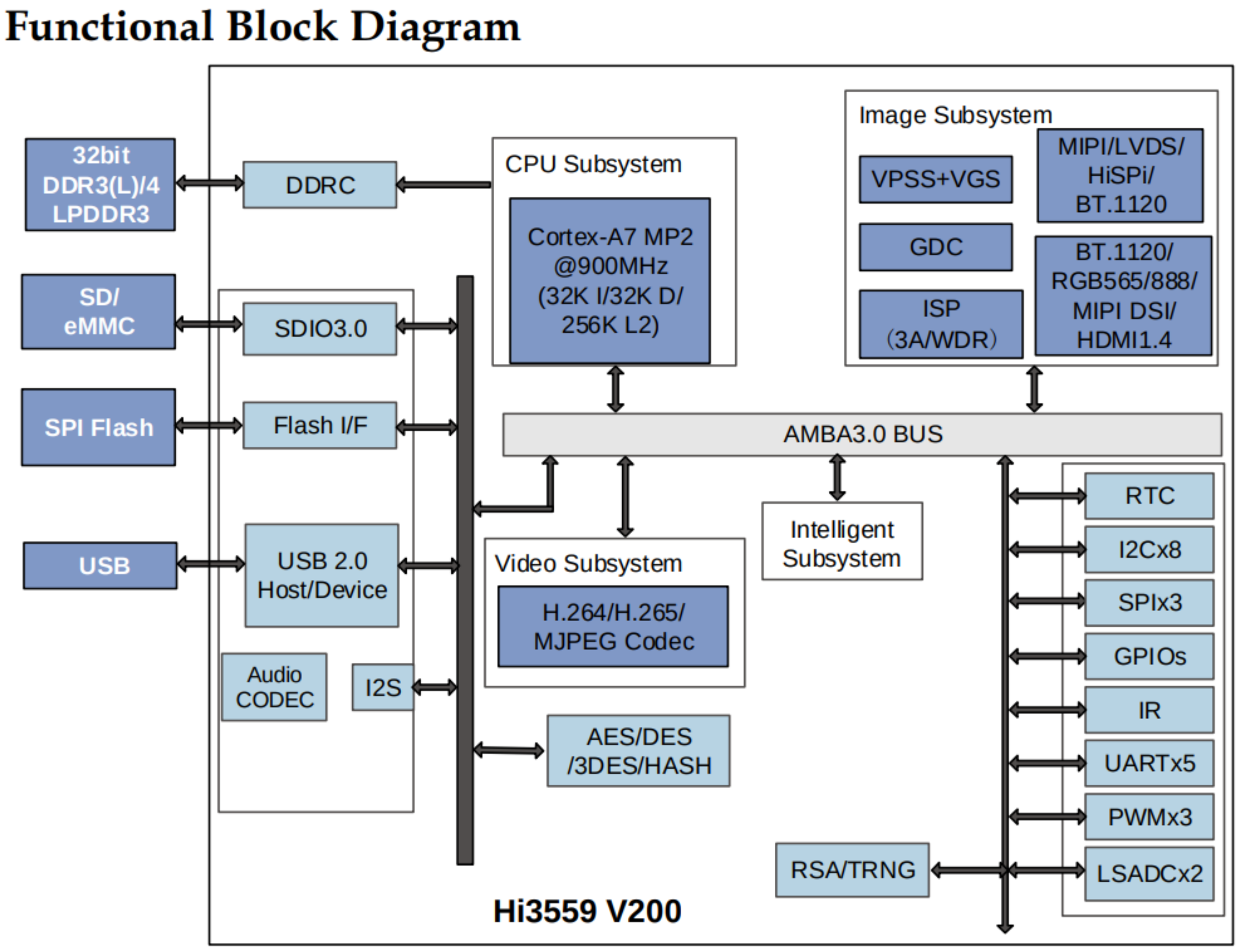

Figure 8: Block Diagram for Hi3559

5.3.2 OmniVision OV798

OV798 is OmniVision made camera soc. It support MIPI interface and H.264 Video encoding up to 720P 60Hz. It can output video via USB 2.0 bus.[11]

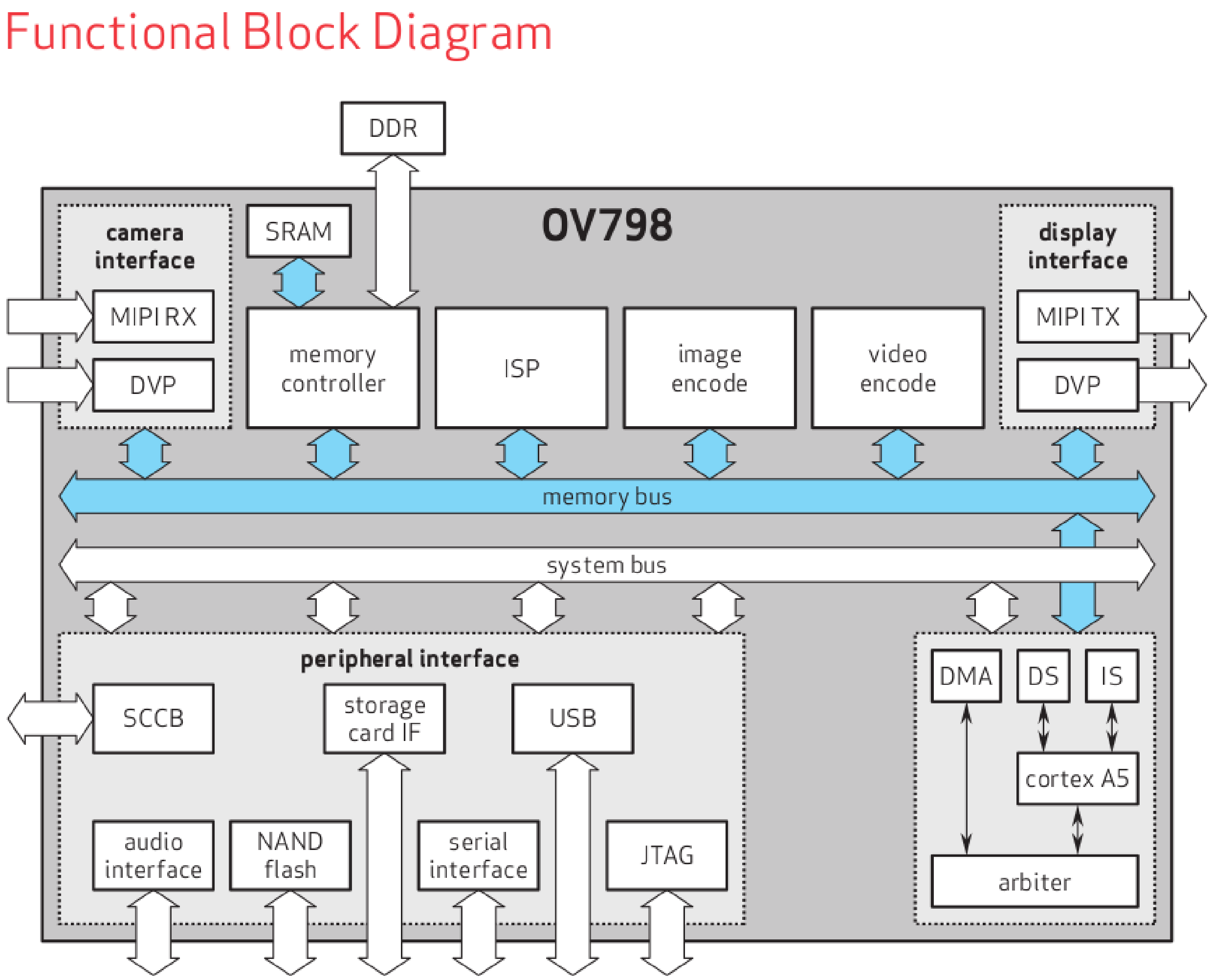

Figure 9: Block Diagram for OV798

5.3.3 Ambarella H22

The Ambarella H22 SoC for consumer applications is a system-on-chip that integrates an advanced image sensor pipeline (ISP), H.265 (HEVC) and H.264 (AVC) encoders , and a powerful Quad core ARM® Cortex™-A53 CPU.[12][5]

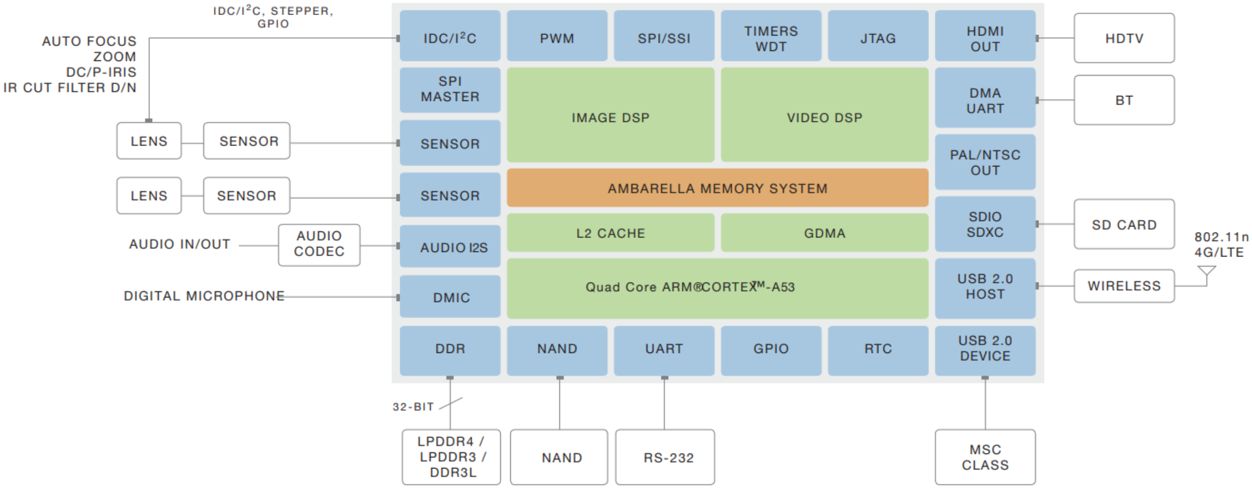

Figure 10: Block Diagram for Ambarella H22

[5] H22 Video SoC for Consumer Applications

5.4 DSP Chip

5.4.1 TMS320C6678 DSP

The TMS320C6678 DSP is a highest-performance fixed/floating-point DSP that is based on TI's KeyStone multicore architecture. Incorporating the new and innovative C66x DSP core, this device can run at a core speed of up to 1.4 GHz.[13][6]

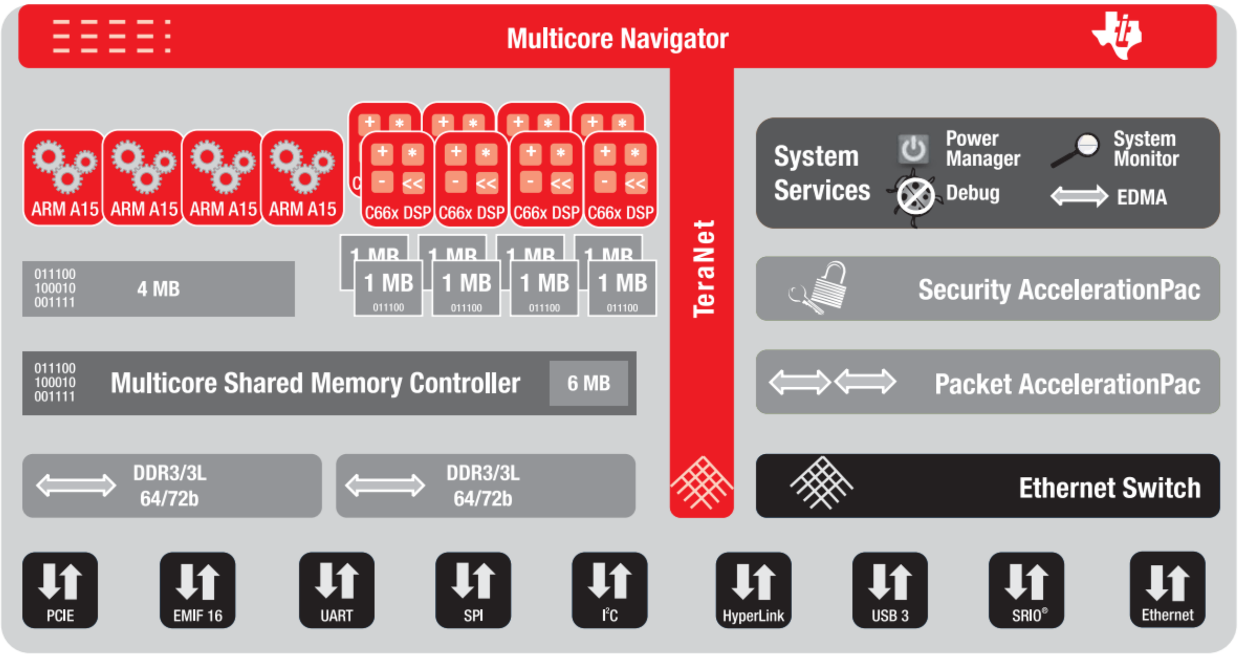

Figure 11: Block Diagram for Ambarella H22

[6] TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor

5.5 GPU Module

5.5.1 Jetson Nano

The Jetson Nano is a GPU computing module what NVIDIA produce for GPU AI computer. It come with 6 MIPI CSI-2 interface and able to handle about 4K 60fps encoding. In the output option, it support PCIe output which can use as the interface. The only problem is it will require extra thermal solution due to the heat it produce and FPGA adoption program to translate the data to backplane bus.[14]

6 System Diagram

6.1 USB camera with SoC FPGA chip

This plan will use USB 2.0 UVC interface to connect one camera to a SoC FPGA chip. The chip will translate the data to back-plane bus and send to the storage device. It can use the builtin USB controller and system driver, which reduce the development time. It could up grade to better camera set by using self developed module. But the builtin USB interface will limited the bit-rate encoding use and might cause blur when the sensor meet frequently vibration.

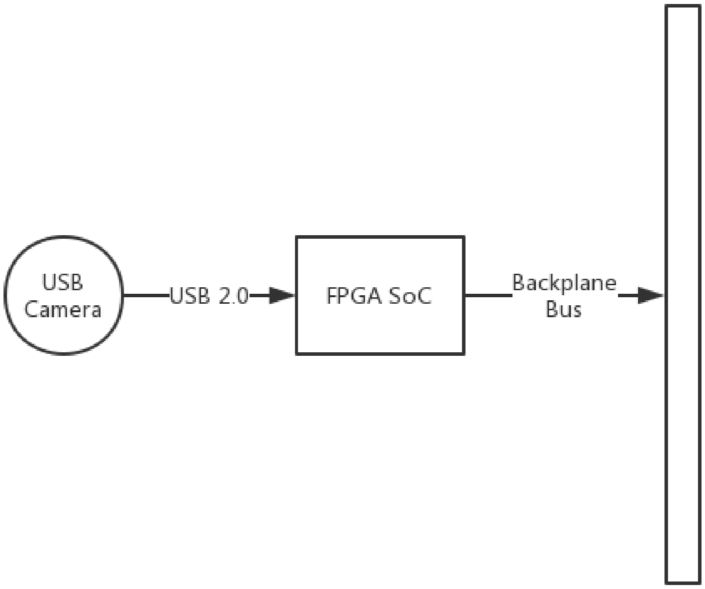

Figure 12: System Diagram for Plan A

Pro	Con
Easy for implement	Low Bitrate
Easy to find sensor	
Come with video encoding	

Table 4: The Pros and Cons Summary

6.2 MIPI camera + CrossLink with FPGA chip

This plan will use a CrossLink chip to translate the MIPI signal to LVDS signal. The FPGA chip will receive the LVDS signal and use internal encoding IP core to process the data and send it out via the backplane bus. The CrossLink and the main chip are both FPGAs. It mean this system can easily increase the number of the camera. Four lane MIPI support up to 4K 180fps frame rate. It also allow future upgrade without change the hardware design due to the flexiblity of FPGA chip.

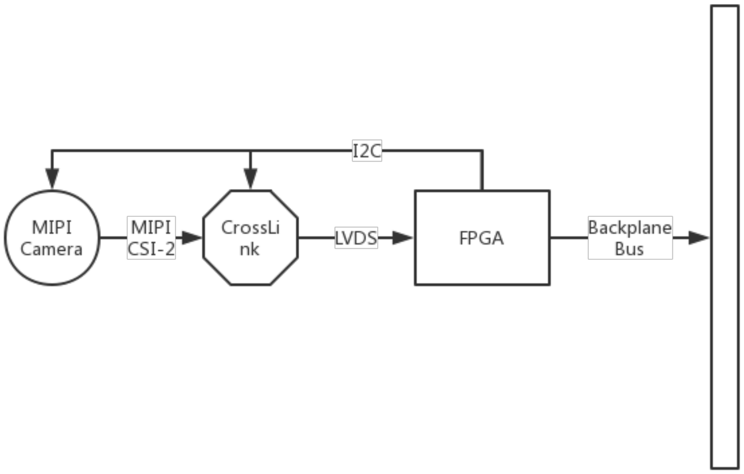

Figure 13: System Diagram for Plan B

Pro	Con
Can use any raspberry Pi Camera	Need Encoding Core
High Speed Sensor(Up to 180 FPS)	Hard in PCB design(100Ω impedance)
Allow futrue upgrade	Long development time
Prevent common mode interference	Need Extra chip for translation

Table 5: The Pros and Cons Summary

6.3 LVDS camera with FPGA chip

This plan will connect a LVDS camera to a FPGA. The FPGA will encode the video and send it via the backplane. The video will be encoded in the FPGA IP core and pack it into the backplane bus packet. Because the camera sensor use the FPGA support standard, it will reduce the problem of format translation. It also allow future upgrade without change the hardware design due to the flexiblity of FPGA chip.

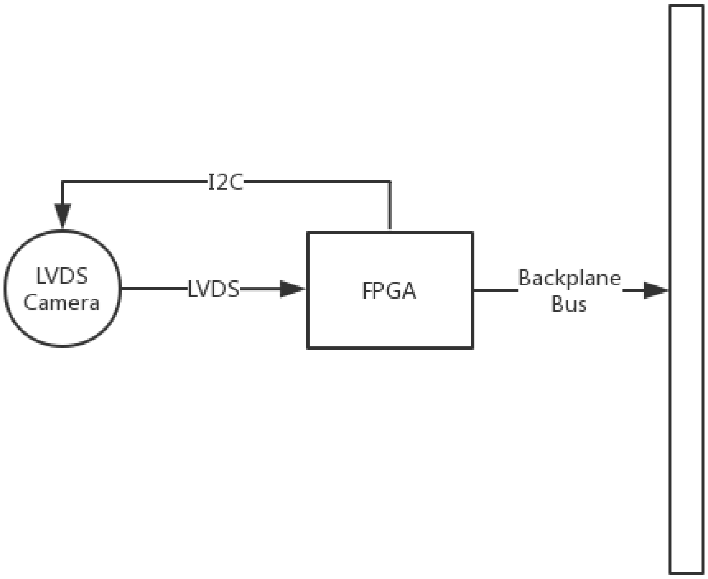

Figure 14: System Diagram for Plan C

Pro	Con
Do not need Extra chip for translation	Need Encoding Core
Cheap sensor	Hard in PCB design(100Ω impedance)
Allow futrue upgrade	Long development time
Prevent common mode interference	

Table 6: The Pros and Cons Summary

6.4 MIPI camera with Multimedia SoC chip

This plan will use the solution that company provide and add additional chip for the data translation from the USB to backplane bus. Because of the solution mostly will be provided by the company, it will be more stable. However, it lose the ability for upgrade. Also, because it use the MIPI bus, it require the PCB manufacture to use 100Ω impedance trace to keep the differential pairs stable.

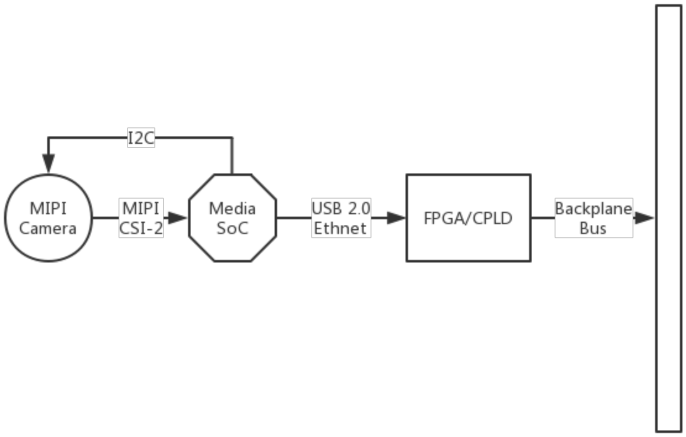

Figure 15: System Diagram for Plan D

Pro	Con
Do not need Extra chip for translation	Only support USB 2.0
Cheap sensor	Hard in PCB design(100Ω impedance)
Do not need Encoding Core	Long development time
Use existing resolution	**Need selecting chip**

Table 7: The Pros and Cons Summary

6.5 MIPI camera with Multimedia DSP and FPGA chip

This plan use DSP chip as the encoder. However the DSP do not have any video interface, it will need FPGA to finish the format translation and feed it into the DSP chip. Also, because it use the MIPI bus, it require the PCB manufacture to use 100Ω impedance trace to keep the differential pairs stable. The DSP itself also have a set of SDK need to be take care off. It might require additional time to construct the program.

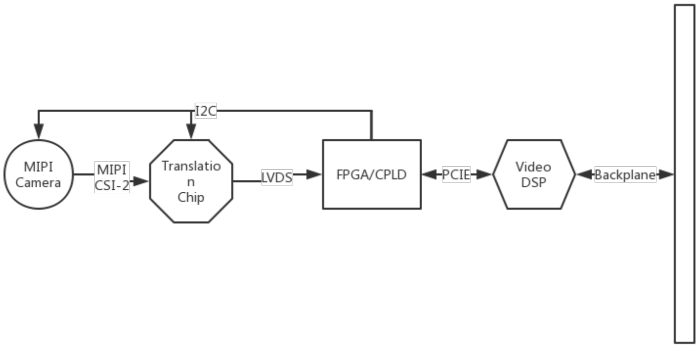

Figure 16: System Diagram for Plan F

Pro	Con
Do not need Encoding Core	Need extra chip for translation and encoding
Use existing resolution	Hard in PCB design(100Ω impedance)
	Long development time

Table 8: The Pros and Cons Summary

7 Bibliography

[1] zhidao, *How to calculate size of video*, 2019. [Online]. Available: https://zhidao.baidu.com/question/175382499941116684.html.

[2] zqh2007, *DVP sensor*, 2019. [Online]. Available: https://blog.csdn.net/zqh2007/article/details/41675189.

[3] LinuxArmbiggod, *LVDS*, 2019. [Online]. Available: https://blog.csdn.net/LinuxArmbiggod/article/details/83863479.

[4] zhang_danf, *MIPI*, 2019. [Online]. Available: https://blog.csdn.net/zhang_danf/article/details/78705133.

[5] bobuddy, *UVC*, 2019. [Online]. Available: https://blog.csdn.net/u010783226/article/details/79176656.

[6] Lattice, *CrossLink: Video Bridging & Processing Optimized*, 2019. [Online]. Available: http://www.latticesemi.com/zh-CN/Products/FPGAandCPLD/CrossLink.aspx.

[7] toshiba, *TC358746AXBG*, 2019. [Online]. Available: https://toshiba-semicon-storage.com/cn/product/assp/interface-bridge/camera-interface.html.

[8] soctechnologies, *MPEG Video CODEC IP Cores*, 2019. [Online]. Available: https://www.soctechnologies.com/ip-cores.

[9] a2etechnologies, *VIDEO COMPRESSION FPGA CORES*, 2019. [Online]. Available: http://www.a2etechnologies.com/products_overview.html.

[10] u010299133, *Hi3559V100*, 2019. [Online]. Available: https://blog.csdn.net/u010299133/article/details/89471231.

[11] OmniVision, *OV798*, 2019. [Online]. Available: https://www.ovt.com/sensors/ov798.

[12] Ambarella, *H22 Video SoC for Consumer Applications*, 2019. [Online]. Available: https://3vpstm1hc6e52739x31131p2-wpengine.netdna-ssl.com/wp-content/uploads/H22-Product-Brief.pdf.

[13] TexasInstruments, *TMS320C6678*, 2019. [Online]. Available: http://www.ti.com/cn/lit/ds/symlink/tms320c6678.pdf.

[14] nVidia, *Jetson Nano*, 2019. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.