OV4688 Camera Module Application Notes

Confidential For Ebensz Only

Last Modified May 6th, 2014

Document Revision: 1.04

OmniVision Technologies, Inc. reserves the right to make changes without further notice to any product herein to improve reliability, function or design. OmniVision does not assume any liability arising out of the application or use of any project, circuit described herein; neither does it convey any license under its patent nor the right of others.

Sensor datasheet is the official document of OmniVision. Software/hardware/dual camera applicaton notes are application guide lines for reference. If there are any difference between sensor datasheet and application notes, please follow sensor datasheet and kindly report the difference to OVT FAE.

This document contains information of a proprietary nature. None of this information shall be divulged to persons other than OmniVision Technologies, Inc. employee authorized by the nature of their duties to receive such information, or individuals or organizations authorized by OmniVision Technologies, Inc.

Table of Contents

Related Documents....4
1. Overview of OV4688 Applications....5
2. Hardware Design of OV4688 Camera Module....6
2.1 Camera Module with 4 lane MIPI Interface....6
2.1.1 Description of Camera Interface....7
2.1.2 PCB/FPCB Layout Guide....8
2.2 Camera Module with 2 lane MIPI Interface....8
2.3 Deal with Lens....9
2.3.1 Light fall off....9
2.3.2 F# of Lens....9
2.3.3 Focus Accuracy at infinite....9
2.3.4 Optical contrast....9
2.4 Calibration of Camera Module....9
2.4.1 White Balance Calibration....9
2.4.2 VCM setting Calibration....9
2.5 Test of OV4688 Camera Module....9
3. Hardware Design with OV4688 Camera....11
3.1 Host Interface....11
3.1.1 Power Supplies....11
3.1.2 Clock....14
3.1.3 Control Signals....14
3.1.4 Image Data Signals....14
3.1.5 Interface with LED driver....14
3.1.6 PCB layout Guide of Host....15
3.2 Mechanical Design....16
3.2.1 Lens Cover....16
3.2.2 Block Wide Angle Light....16
3.2.3 Keep Camera Module Cool....16
4. Hardware Operation of OV4688 Camera Module....18
4.1 Cut off all Powers when Camera is not used....18
4.1.1 System Power On....18
4.1.2 Enter Camera....19
4.1.3 Switch to Other Camera....21
4.1.4 Switch from Other Camera....27
4.1.5 Exit Camera....28
4.1.6 System Power Off....29
4.2 Cut off DVDD When Camera is not used....30
4.2.1 System Power On....30
4.2.2 Enter Camera....31
4.2.3 Switch to Other Camera....32
4.2.4 Switch from Other Camera....32
4.2.5 Exit Camera....33
4.2.6 System Power Off....34
5. Software Operation of OV4688....35
5.1 Register Settings for 4 Lane MIPI, 1008Mbps....35
5.1.1 Initialization (Global Setting)....35
5.1.2 Full Resolution 2688x1520 90 fps....43

Confidential For Ebensz Only

5.1.3 2016x1520 90 fps....44
5.1.4 EIS 1080p 2120x1196 120fps....44
5.1.5 EIS 720p 1344x760 180fps....45
5.2 Register Setting for 4 Lane MIPI, 672Mbps....46
5.2.1 Initialization(Global Setting)....46
5.2.1 Full Resolution 2688x1520 60fps....52
5.2.2 2016x1520 60fps....52
5.2.3 EIS 1080p 2120x1196 75fps....53
5.2.4 EIS 720p 1344x760 110fps....54
5.3 Settings for 2 Lane MIPI, 672Mbps....55
5.3.1 Initialization (Global Settings)....55
5.3.2 Full Resolution 2688x1520 30fps....60
5.3.3 2016x1520 30fps....61
5.3.4 EIS 1080p 2120x1196 30fps....62
5.3.5 EIS 720p 1344x760 58.5fps....62
5.4 Drive capability....63
5.5 I/O Control....63
5.6 MIPI Stream Control....64
5.7 Mirror and Flip....64
5.8 BLC (Black Level Calibration) Strategy....65
5.9 Load Calibration Data....67
5.2.1 Calibration Data Assignment....67
5.2.1 Load Calibration Data....73
5.3 Driver Functions....80
5.3.1 Exposure....80
5.3.2 Gain....81
5.3.3 Capture Sequence....82
5.3.5 SCCB Group Write....83
5.3.6 Capture with Flash LED....84
5.4 Verification of Camera Driver Functions....84
6. Tuning Image Quality....85
6.1 Using Calibration Data....85
6.2 Find Typical Camera Module and Corner Camera Modules....85
6.3 Tune with ISP Tools....85
6.4 Verify Settings Tuned....85
Revision History....86

Confidential For Ebensz Only

Related Documents

"OV4688 Datasheet"

"OV4688 Camera Module Calibration"

"Camera Module Image Quality Evaluation Procedure and Criterion – RAW Sensor"

"Camera Analyzer User Guide"

Confidential For Ebensz Only

1. Overview of OV4688 Applications

OV4688 is a 1/3 inch ultra performance 4M camera supporting 4 lane MIPI interface. It can be used as

Ultra Performance 4M main camera with 4 lane MIPI interface

Ultra Performance 4M sub camera with 4/2 lane MIPI interface

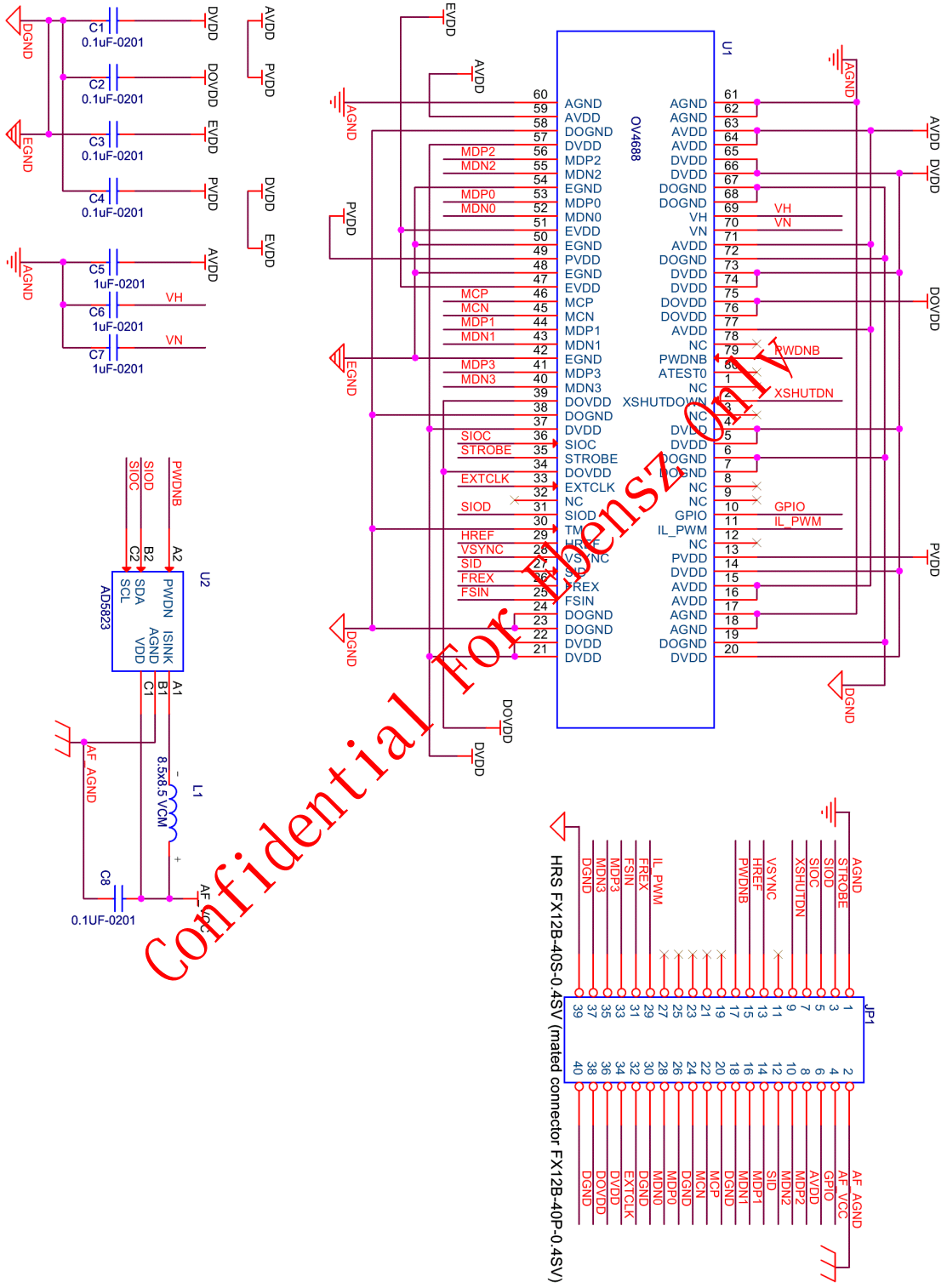
High Performance 1080p video camera with 2 lane MIPI interface

High Performance 720p video camera with 2 lane MIPI interface

Confidential For Ebensz Only

2. Hardware Design of OV4688 Camera Module

2.1 Camera Module with 4 lane MIPI Interface



Confidential For Lensz only

2.1.1 Description of Camera Interface

Table – 1 Description of 4 Lane MIPI Camera Interface

Pin #	Pin Name	Explanation
1	AGND	Analog ground.
2, 4	AF_AGND and AF_VCC	power supply for built-in VCM driver. Although AF_VCC is 2.8~3.3V, it is strongly recommended to use 3.3V for better auto focus performance.
3, 29	STROBE and IL_PWM	flash LED control interface. It could be used to control LED driver. Leave it open if not used.
5	SIOD	data of SCCB(Serial Camera Control Bus). It should be pulled to DOVDD with pull up resistors. Typical value of pull up resistors are 4.7K.
6	GPIO	General Purpose I/O. Leave it open if not used.
7	SIOC	clock of SCCB(Serial Camera Control Bus). It should be pulled to DOVDD with pull up resistors. Typical value of pull up resistors are 4.7K.
8	AVDD	2.6~3.0V of sensor analog power (clean).
9	XSHUTDOWN	active low, it should be connected to DOVDD outside of module if unused.
10, 12	MDP2 and MDN2	MIPI 3rd data lane positive and negative output.
13	VSYNC	Video output vertical signal. Leave it open if not used.
14	SID	SCCB address select 0: SCCB address = 0x6c after reset 1: SCCB address = 0x20 after reset OV4688 has another SCCB address 0x42 which doesn't related to SID.
15	HREF	Video output horizontal signal, leave it open if not used.
16, 18	MDP1 and MDN1	MIPI 2nd data lane positive and negative output.
17	PWDNB	active low, it should be pulled high to DOVDD outside of module if unused.
20, 26, 32, 39, 40	DGND	Digital ground. AGND and DGND should be separated and connected to a single point outside camera module (Do not be connected inside module).
22, 24	MCP and MCN	MIPI clock lane positive and negative output.

Confidential For Ebensz Only

Pin #	Pin Name	Explanation
26, 28	MDP0 and MDN0	MIPI 1st data lane positive and negative output.
33	FSIN	Frame sync input, leave it open if not used
34	XVCLK	external clock input.
35, 37	MDP3 and MDN3	MIPI 4th data lane positive and negative output.
36	DVDD	1.1V~1.3V of sensor core power.
38	DOVDD	1.7~3.0V of sensor digital IO power (clean). 1.8V is recommended.

2.1.2 PCB/FPCB Layout Guide

1. Sensor AGND and DGND should be separated and connected to a single point outside PCB (Do not be connected inside module).

2. EVDD/EGND are power/ground for MIPI core. DVDD and EVDD should be two separated nets, and only connected at a single point.

3. Capacitors should be close to the related sensor pins.

4. If more space available, use 1uF capacitor between DVDD and DGND.

5. The traces of MCP, MCN, MDPx, and MDNx should have the same or similar length. The impedance transmission lines should be controlled at 100 Ohm. The differential pairs should be separated from other wires at least 3 space(no matter in same layer or opposite layer) or with guarded ground wires. Try to minimize number of vias on differential pairs and keep number of vias same for all differential pairs.

6. The traces from AF_VCC to L1+ and L1- to pin VCMISNK, and the trace of AF_AGND need to be paid extra attention because of high electrical current.

7. AVDD and DOVDD should be separated from clock wires as XVCLK, SCL, SDA, no matter in same layer or in opposite layers.

8. XVCLK should be close to GND and keep away from other signals.

9. The number of vias on ground path should be as less as possible.

Confidential For Lensz Only

2.2 Camera Module with 2 lane MIPI Interface

Camera with 4 lane MIPI interface could be used with 2 lane MIPI interface only, just leaving MDP2, MDN2, MDP3 and MDP3 open.

2.3 Deal with Lens

2.3.1 Light fall off

Light fall off means the corner of image is darker than center of image. It is caused by CRA of image sensor and CRA of lens. The RI (relative illuminant) is a index to measure light fall off of lens.

$$RI = \frac{corner\ illuminant}{center\ illuminant}$$

The lens shading correction function of OV4688 could be turned on to apply extra gain to compensate the corner brightness and make the whole picture looks same bright. The extra gain cause extra noise at corners of image. So actually RI reflects corner noise at bright condition. To reduce corner noise at bright condition, it is recommended to select lens with higher RI.

2.3.2 F# of Lens

F# of lens indicates that how many light is pass to sensor through lens. To get better low light sensitivity, more light is required to pass through lens. So select a lens with bigger aperture and smaller F# is required. For OV4688, F2.2 and F2.0 lens are recommended.

2.3.3 Focus Accuracy at infinite

The focus of auto focus module is quite different as fixed focus module. Fixed focus module generally focus at some distance and use hyper-focus to cover infinite. Auto focus module need to focus infinite exactly when VCM is not driven to get perfect landscape pictures.

2.3.4 Optical contrast

The optical contrast of lens is very important to picture quality. If the optical contrast of lens is not good, the picture would looks foggy. Though it could be improved by increase the sensor contrast to make the picture sharper, the higher sensor contrast would make the detail lost of dark area of the picture. To get better optical contrast, select lens with higher MTF and lower flare is strongly recommended. Also the wide angle light (wider than FOV of lens) should be carefully blocked in phone design.

2.4 Calibration of Camera Module

2.4.1 White Balance Calibration

Auto white balance algorithm of ISP need detect color temperature of light by sensor response. Sensor response to color temperature of light varies from one sensor to another because of semiconductor process variation. To minimize sensor variation to color temperature, white balance calibration could be done at camera module level and store calibration information in OTP of OV4688. Then the calibration information could be loaded from OTP and applied to ISP.

2.4.2 VCM setting Calibration

The lens position vs VCM setting also varies from one camera module to another. It could be calibrated and stored in OTP of OV4688.

For detail information of OV4688 camera module calibration and how to use calibration data, please read document “OV4688 Camera Module Calibration.pdf”.

2.5 Test of OV4688 Camera Module

Since OV4688 is a raw sensor, the final performance of OV4688 camera module can not be shown at camera module level. Test of OV4688 camera module is quite different than test of SOC

Confidential For Ebensz Only

camera modules. Some test items of SOC sensors as AWB, color reproduction, SNR could not be tested camera camera module level. For detail information of raw camera module test, please read document “Camera Module Image Quality Evaluation Procedure and Criterion – RAW Sensor”. Test items of OV4688 camera module includes:

Resolution Test

Test resolution with ISO 12233 chart.

Auto Focus Test

Test resolution at distances as >3 meter, full FOV of 4x ISO 12233 chart, close up < 10cm.

Shading Test

Test luminance shading and color shading.

Blemish Test

Check blemish of camera modules.

Confidential For Ebensz Only

3. Hardware Design with OV4688 Camera

3.1 Host Interface

3.1.1 Power Supplies

Host provide 4 powers to OV4688 camera module. TAVDD is 2.8V clean analog power. DOVDD is 1.8V digital I/O power, DVDD is 1.2V digital core power. Power supply capability should match with power consumption of the target. Normally we recommend power supply maximum output current is 2~3 times maximum power consumption of the target.

Table – 3 Power supply requirement

Name	Voltage requirements	Max current requirements
AVDD	2.8V (2.6V~3.0V)	45mA
AF_VCC	2.8V(2.8V~3.3V)	100mA
DOVDD	1.8V(1.7V~3.0V)	7mA
DVDD	1.2V(1.1V~1.3V)	150mA

Powers of OV4688 could be supplied by PMU or regulator. Figure 3 shows powers supplied by PMU. Figure 4 show powers supplied by regulators.

Confidential For Ebensz Only

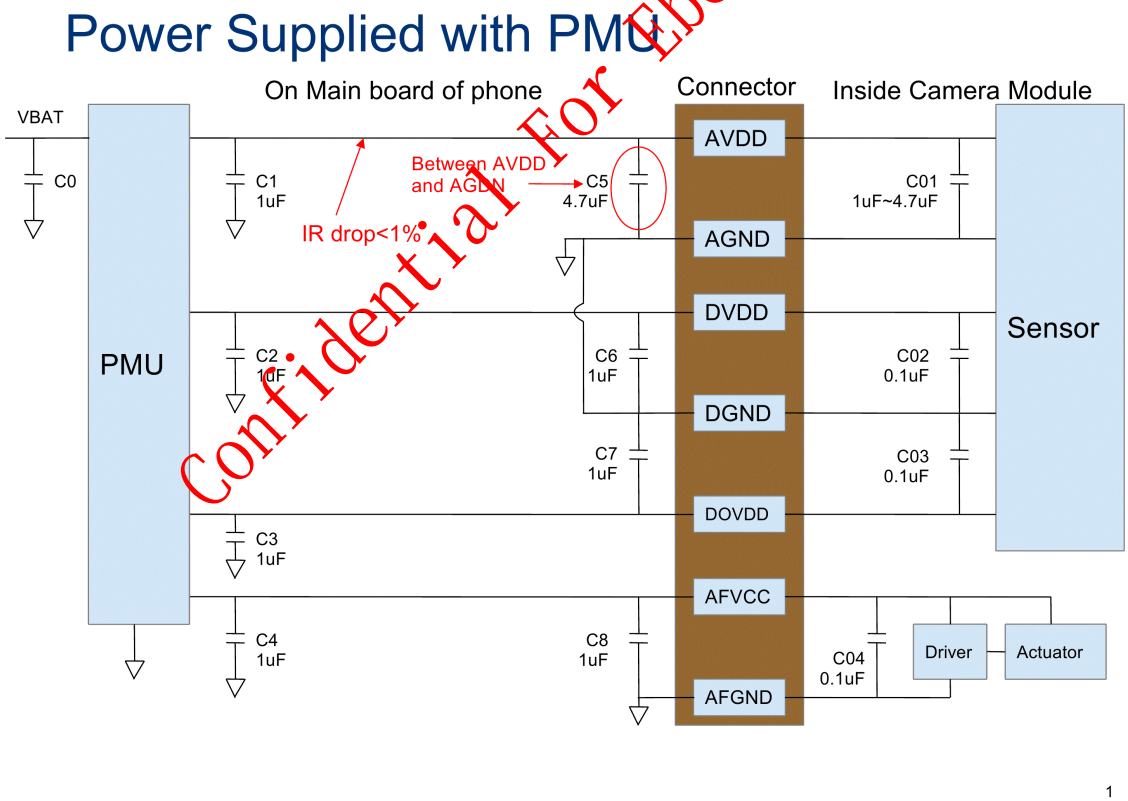


Figure 3 – Powers supplied by PMU

Output Capacitor of PMU

As PMU is placed far away from camera, output capacitor of PMU should split to 2 capacitors, one close to PMU(as C1, C2, C3, C4 in Figure 3), another close to connector of camera(as C5, C6,

C7, C8 in Figure 3).

The cap close to connector on AVDD (C5 >=4.7) shall be placed between AVDD and AGND, and before AGND connects to system ground panel.

Power Trace

Less than 1% IR drop is required to minimize system power related H-noise. With assumption copper trace 1 oz/ft^2, and 100mA peak current on power, the minimum power trace width/length shall be 1:400. If trace length is 3 inches, the power trace needs be 7.5mil. If peak power is 200mA, the width/length ratio shall be 1:200 output capacitor of regulator should be placed close to connector of camera.

If customer is using ground plane for system GND, AVDD must be on adjacent layer for best return current reference, and there should be no active signals, such as clock, adjacent to AVDD trace.

Ground

AGND and DGND area shall be as big as possible – preferably an area for AGND.

AGND and DGND are separated inside module, connected together close to the connector to system GND layer from single point

Ground of capacitors of AVDD, DOVDD and DVDD close to connector should be connected to camera ground first, then connected to system ground

Confidential For Ebensz Only

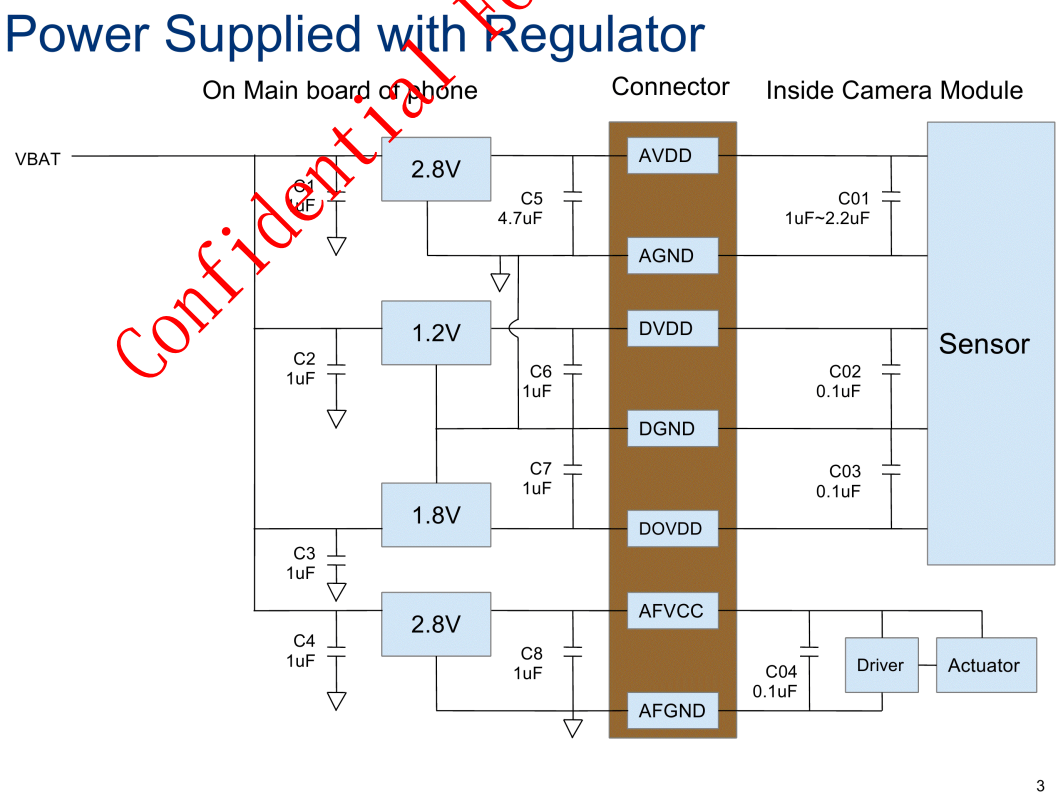


Figure 4 – Powers supplied by regultor

Regulator

Regulators should be placed close to the connector of camera module.

Output capacitor of regulator shall be placed close to connector of camera.

PSRR and Load Transient Response are very important for AVDD regulator.

Ground

AGND and DGND area shall be as big as possible – preferably an area for AGND and DGND.

AGND and DGND are separated inside module, connected together close to the connector to system GND layer from single point.

Ground of capacitors of AVDD, DOVDD and DVDD close to the connector should be connected to camera ground first, then connected to system ground

PSRR(Power Supply Ripple Rejection) of Regulator

PSRR is a measure of how much ripple at input of regulator could be transferred to output of regulator. Regulator may have low PSRR at mid frequency and high frequency. Regulator of AVDD requires PSRR as high as possible.

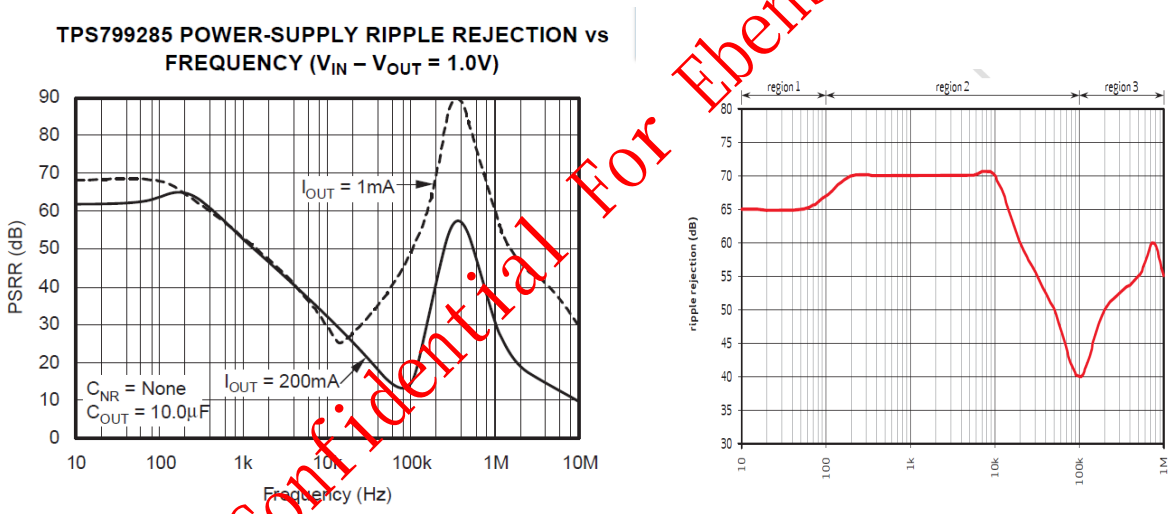
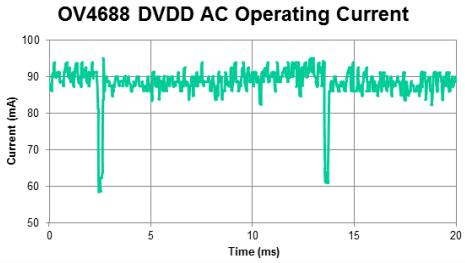


Figure 5 – Regulator(left) with poor PSRR and regulator(right) with good PSRR

Load Transient Response of Regulator

Due to working current changes at horizontal active and blanking, and currrent change may cause voltag change, sensor DVDD, DOVDD and AVDD may be noise source.



Confidential For Ebensz only

Load transient response shows how much voltage changes when current changes for a given value. The less voltage change, the better.

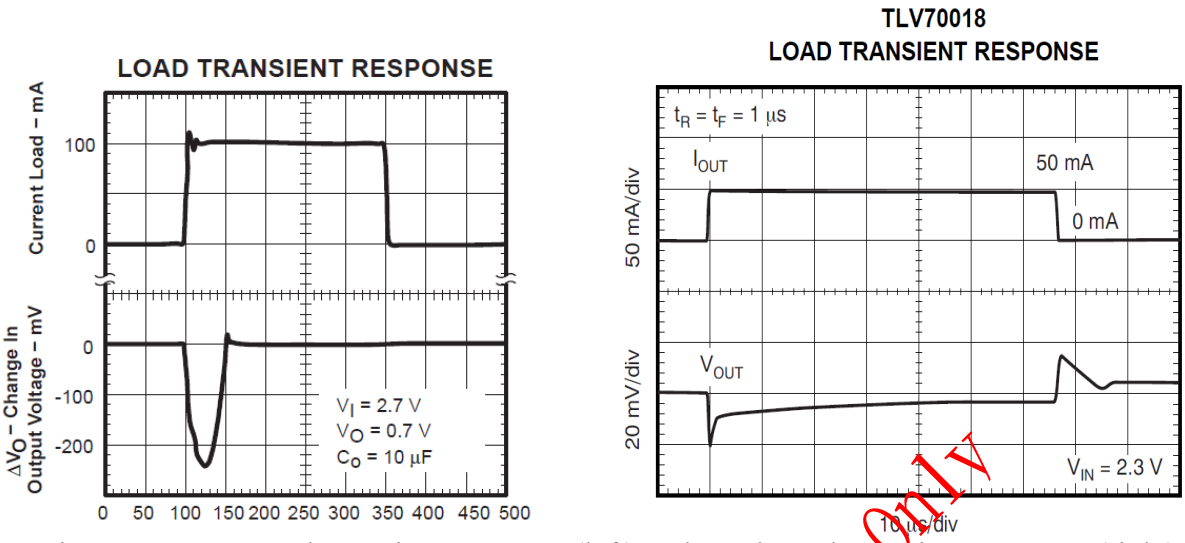


Figure 6 – Poor Load Transient Response(left) and good Load Transient Response(right)

Confidential For Hensz Only

3.1.2 Clock

Clock frequency of 6Mhz to 64Mhz could be supplied to camera module. 24Mhz is recommended. If other clock frequency is used, please contact with OmniVision local FAE to adjust register settings of OV4688. The rising and falling edge of clock should be less than 27% of whole clock period.

3.1.3 Control Signals

SIOC/SIOD, from host to OV4688. I²C bus control signals. Control OV4688 registers. Slave ID of OV4688 is controlled by SID pin. If SID is low, the slave ID of OV4688 is 0x6c. If SID is high, the slave ID of OV4688 is 0x20. OV4688 has another slave ID comes from register 0x3006, which doesn't related to status of pin SID and has value of 0x42.

XSHUTDOWN active low, from host to OV4688. Set OV4688 to standby mode (register values are not kept) when XSHUTDOWN is low.

PWDNB active low, from host to OV4688. Set OV468888 to power down mode (register values are kept) when PWDNB is low.

STROBE and IL_PWM, from OV4688 to host. Control flash LED.

3.1.4 Image Data Signals

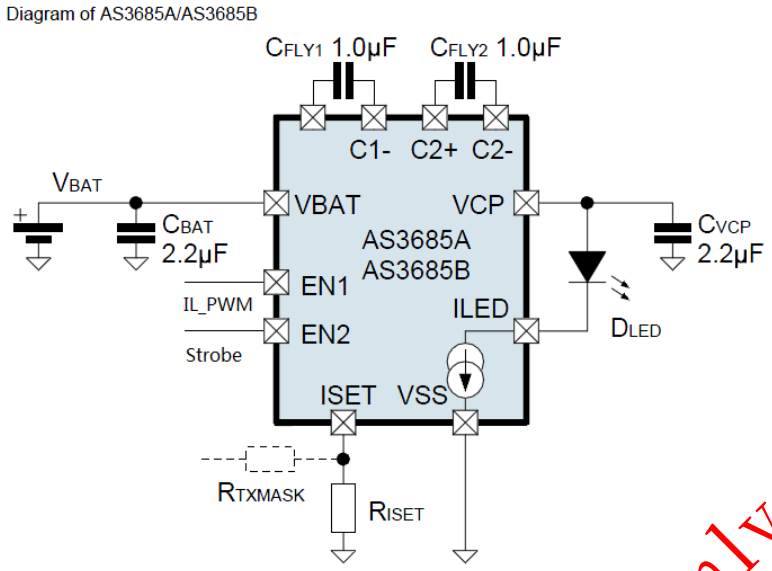
MCP/MCN clock of MIPI.

MDP0/MDN0, MDP1/MDN1, MDP2/MDN2, MDP3/MDN3, 4 lane MIPI interface.

MDP0/MDN0, MDP1/MDN1, 2 lane MIPI interface.

3.1.5 Interface with LED driver

Strobe and Shutter could be used to control LED driver. Strobe is used to sync LED with OV4688 exposure. Shutter could be used as general GPIO to control torch mode and flash mode of LED driver. Below is a sample schematic of OV4688 interface with LED driver – AS3685A.



3.1.6 PCB layout Guide of Host

1. The traces of MCP, MCN, MDPx, and MDNx should have the same or similar length. The impedance transmission lines should be controlled at 100 Ohm. The differential pairs should be separated from other wires at least 3 space(no matter in same layer or opposite layer) or with guarded ground wires. Try to minimize number of vias on differential pairs and keep number of vias same for all differential pairs.

2. The traces of AF_VCC, AVDD and DVDD need to be paid extra attention because of high electrical current.

3. XVCLK should be keep away from other signals.

4. Decouple capacitors of AVDD, DVDD and DOVDD should be closed to connector of camera module.

5. AGND and DGND could be connected together on main PCB. Decouple capacitors should be connected to seperated AGND or DGND before the grounds are connected.

Confidential For Eberisz Only

3.2 Mechanical Design

3.2.1 Lens Cover

The lens cover is the cheapest part in optical path. But it could affect picture quality very much. The lens cover should be made with optical glass with AR coating at both side. Otherwise, the lens cover may cause sensitivity loss and/or stronger lens flare.

3.2.2 Block Wide Angle Light

The wide angle light (light outside FOV of lens) may cause lens flare which lower contrast of image and cause image looks foggy. So the wide angle light need to be blocked by mechanical design.

3.2.3 Keep Camera Module Cool

As OV4688 camera module runs in high frame rate (up to 90fps full resolution), and high resolution with auto focus, the power consumption of 4M camera module is higher than normal 5M camera module. Below are samples of OV4688 working states and typical power consumption.

Table – 4 Power consumption of camera module

OV4688 Output	Mode	AVDD (mA)	DVDD (mA)	DOVDD (mA)	AF_VCC (mA)	Typical Power Consumption (mW)
Full resolution@30fps	Landscape	40	120	3	0	261
	Portrait	40	120	3	30	345
	Macro	40	120	3	80	485
Full resolution@90fps	Landscape	40	120	3	0	261
	Portrait	40	120	3	30	345
	Macro	40	120	3	80	485

Note:

AVDD = 2.8V, DVDD = 1.2V, DOVDD = 1.8V, AF_VCC = 2.8V.

The worst case power consumption of OV4688 camera module is 485mW(Macro mode). The heat generated by the power consumption should be dispatched out of camera module. The following table shows heat conductivity of normal materials.

Confidential For Eberspächer Only

Table – 5 Heat conductivity of material

Material	Heat Conductivity (W/m.K)
Copper	383
Aluminum	204
Steel	47
Stainless Steel	29
FR4	0.8
High heat conduct FR4	2.5
Air	0.02

The recommended heat dispatch solution includes

1. If camera module contacts with PCB, cover the area under camera module with copper. So that the heat from camera module is dispatched to PCB.

2. If camera module contacts with back cover of LCD, please add a thin film of copper on surface of LCD back cover, so that the heat is conducted to LCD cover.

3. Add heat sink to OV4688 camera module and conduct heat from heat sink to metal case of phone.

Confidential For Ebensz Only

4. Hardware Operation of OV4688 Camera Module

The powers of OV4688 could be always supplied or cut off when camera not used. OV4688 standby could be controlled by PWDNB or XSHUTDOWN. So there are 4 combinations of OV4688 hardware operation schemes.

Table – 6 Comparison of hardware operation schemes

Camera off Power Control	Controlled by	Camera off Current	Standby Current	Switch Camera	Comment
Cut off all	XSHUTDOWN	min	min		
	PWDNB	min		fast	Min power consumption for camera off and fast switch
All powers on	XSHUTDOWN	min	min	software	Min power consumption for standby and camear off
	PWDNB			fast	

After compared 4 operation schemes, 2 are recommended for design usage. They are:

1. Minimum power consumption for camera off and fast switch

Cut off all powers when camera not used. XSHUTDOWN connected to DOVDD. PWDNB is used to control operation.

2. Minimum power consumption for standby and camera off

All powers on when camera not used. PWDNB connected to DOVDD. XSHUTDOWN is used to control operation.

Confidential For Ebensz Only

4.1 Cut off all Powers when Camera is not used

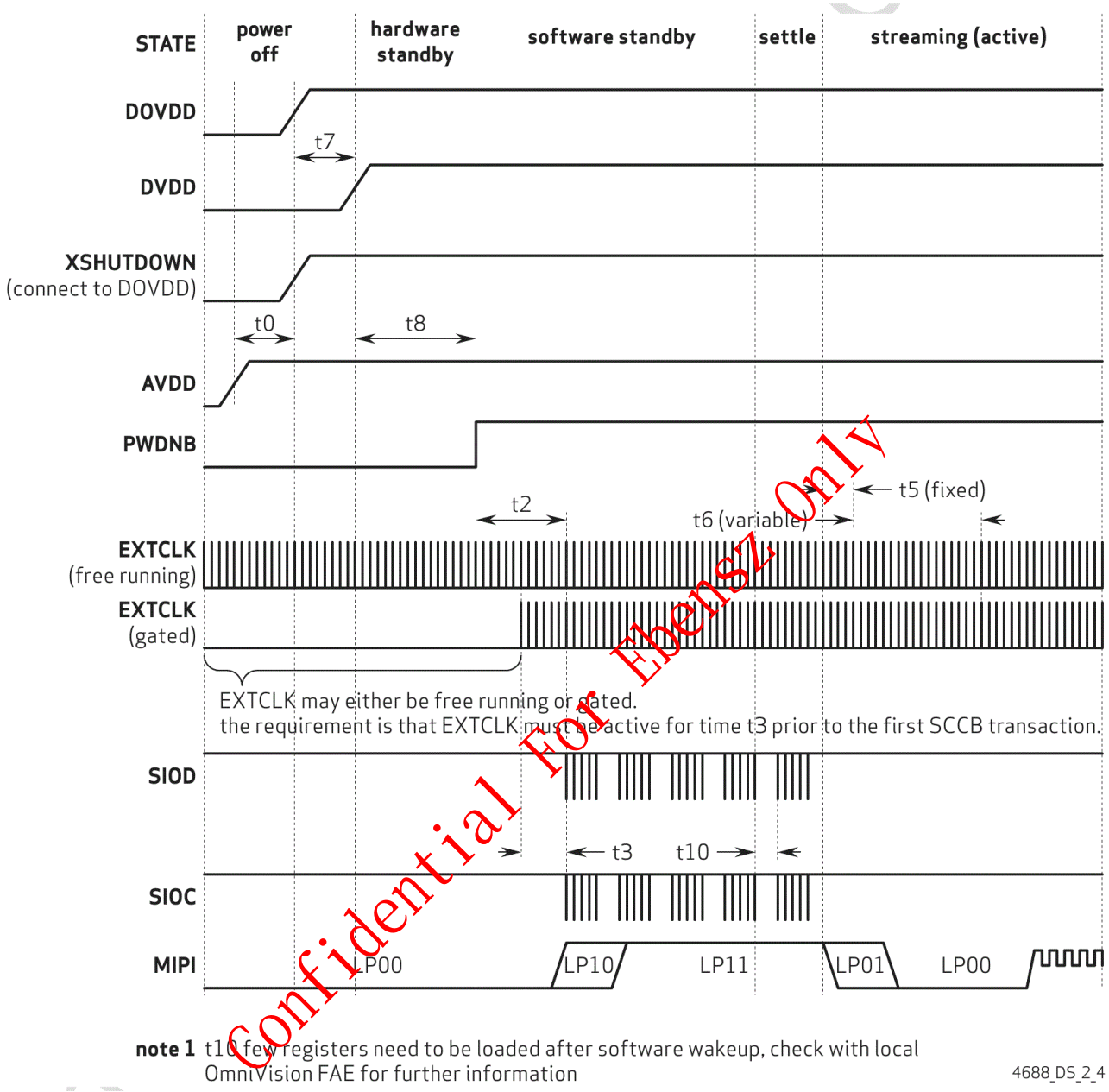
Cut off all powers when camera not used. XSHUTDOWN connected to DOVDD. PWDNB is used to control operation.

Host provide 3 powers to OV4688 camera module, AVDD, DOVDD and DVDD.

4.1.1 System Power On

Keep AVDD, DOVDD and DVDD off. Pull low all input pins to OV4688.

4.1.2 Enter Camera



Confidential For Ebensz Only

1. Power up. Apply AVDD fist. XSHUTDOWN pin should be connected to DOVDD. PWDNB pulled to low.

2. After t0, apply DOVDD (t0>=0ms).

3. After t7, apply DVDD (t7>=0ms).

4. After t8 time after last power applied, PWDNB is pulled high (t8>=0ms).

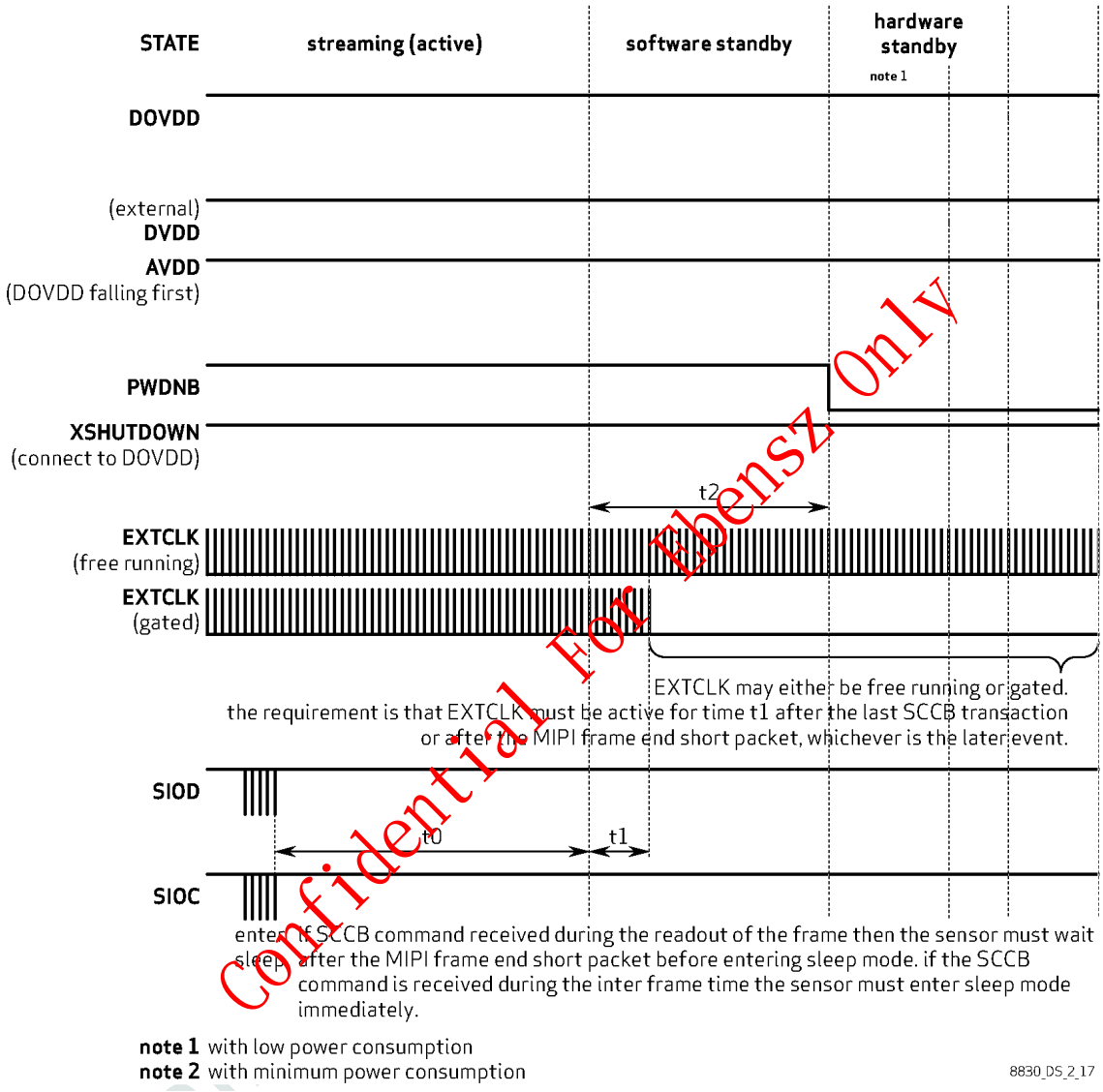
5. XVCLK could be gated or free running. If gated, apply XVCLK 8192 XVCLK cycles(t3) before SCCB access.

6. SCCB could be accessed after 8192(t3) cycle of XVCLK applied and after 8192(t2) cycle of

PWDNB pulled high.

7. Initialize OV4688. Please find initialization settings of OV4688 from section 5.1. After SCCB initialization and video stream turned on, the first video frame come out after 10ms (t5) + exposure time(t6).

4.1.3 Switch to Other Camera



Confidential For Ebensz Only

1. Software standby. Send SCCB registers to set OV4688 to software standby mode.
6c 0100 00

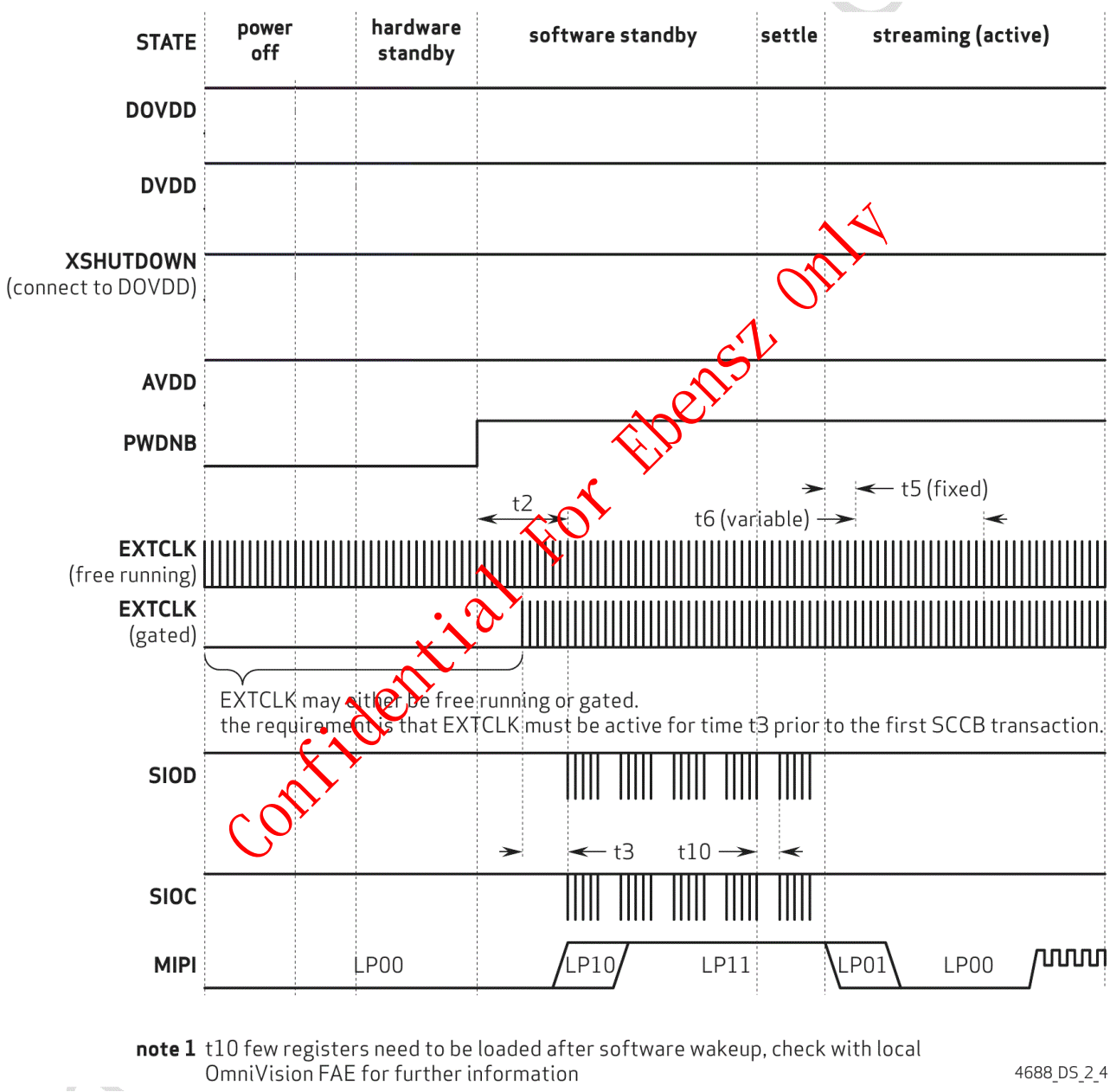
2. Delay 1 frame, then write following registers
6c 301a f9
6c 4805 03
if SCCB command received during the readout of the frame then the sensor must wait after the MIPI frame end short packet before entering sleep mode. If the SCCB command is received during the inter frame time, the sensor enter sleep mode immediately(t0).

3. if XVCLK is gated, pull low XVCLK after 512(t1) cycles of MIPI frame end.

4. After 512(t2) cycles of MIPI frame end, pull low PWDNB, set OV4688 to hardware power down mode.

5. Enter other camera.

4.1.4 Switch from Other Camera



Confidential For Ebensz Only

1. power down other camera.

2. Pull high PWDNB, set OV4688 from hardware power down to software power down.

3. XVCLK could be free running or gated. If gated, apply XVCLK.

4. After 8192 cycles(t3) of XVCLK applied and 8192 cycles(t2) of PWDN pulled high, write SCCB

registers to wake up OV4688 from software power down.

6c 0100 01

6c 301a f9

5. After 10ms write registers

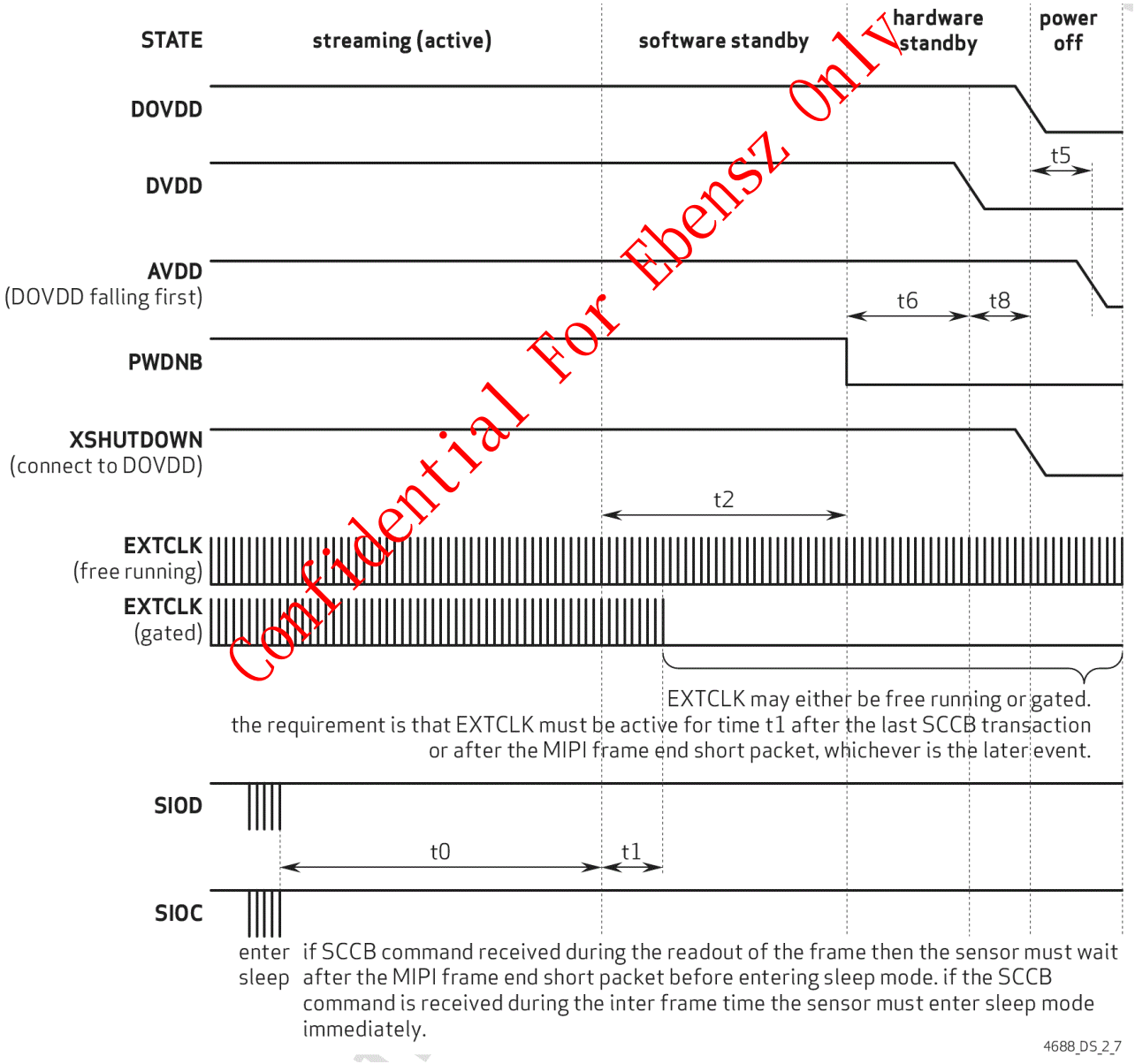
6c 301a f1

6c 4805 00

6c 301a f0

First frame image will output after 10ms(t5) + exposure time(t6).

4.1.5 Exit Camera



Confidential For Ebensz Only

1. Software standby. Send SCCB registers to set OV4688 to software standby mode.

6c 0100 00
2. Delay 1 frame, then write following registers
6c 301a f9
6c 4805 03

if SCCB command received during the readout of the frame then the sensor must wait after the MIPI frame end short packet before entering sleep mode. If the SCCB command is received during the inter frame time, the sensor enter sleep mode immediately(t0).

3. if XVCLK is gated, pull low XVCLK after 512 cycles(t1) of MIPI frame end.

4. After 512 cycles(t2) of MIPI frame end, pull low PWDNB, set OV4688 to hardware power down mode.

5. After t6, Turn off DVDD(t6>=0).
6. After t8, turn off DOVDD(t8>=0).
7. After t5, turn off AVDD(t5>=0).

When OV4688 is powered off, please pull low all input pins (except SCCB pins) to OV4688. If any input pin (except SCCB pins) of OV4688 is pulled high, the high input pin may leak current to DVDD of OV4688 and make the sensor unstable.

4.1.6 System Power Off

No operation for OV4688.

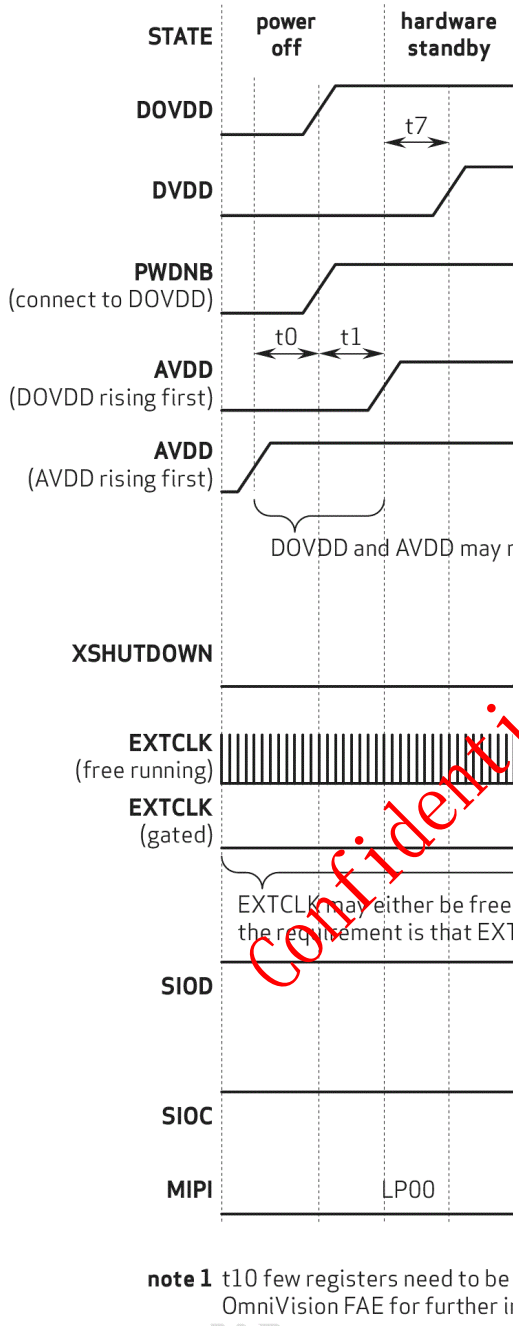
Confidential For Ebensz Only

4.2 Cut off DVDD When Camera is not used.

Cut off DVDD power when camera not used. PWDNB is connected to DOVDD. XSHUTDOWN is used to control operation.

Host provided 3 powers to OV4688, DOVDD, DVDD and AVDD. DVDD is turned off when OV4688 camera is off. DOVDD and AVDD are always on.

4.2.1 System Power On



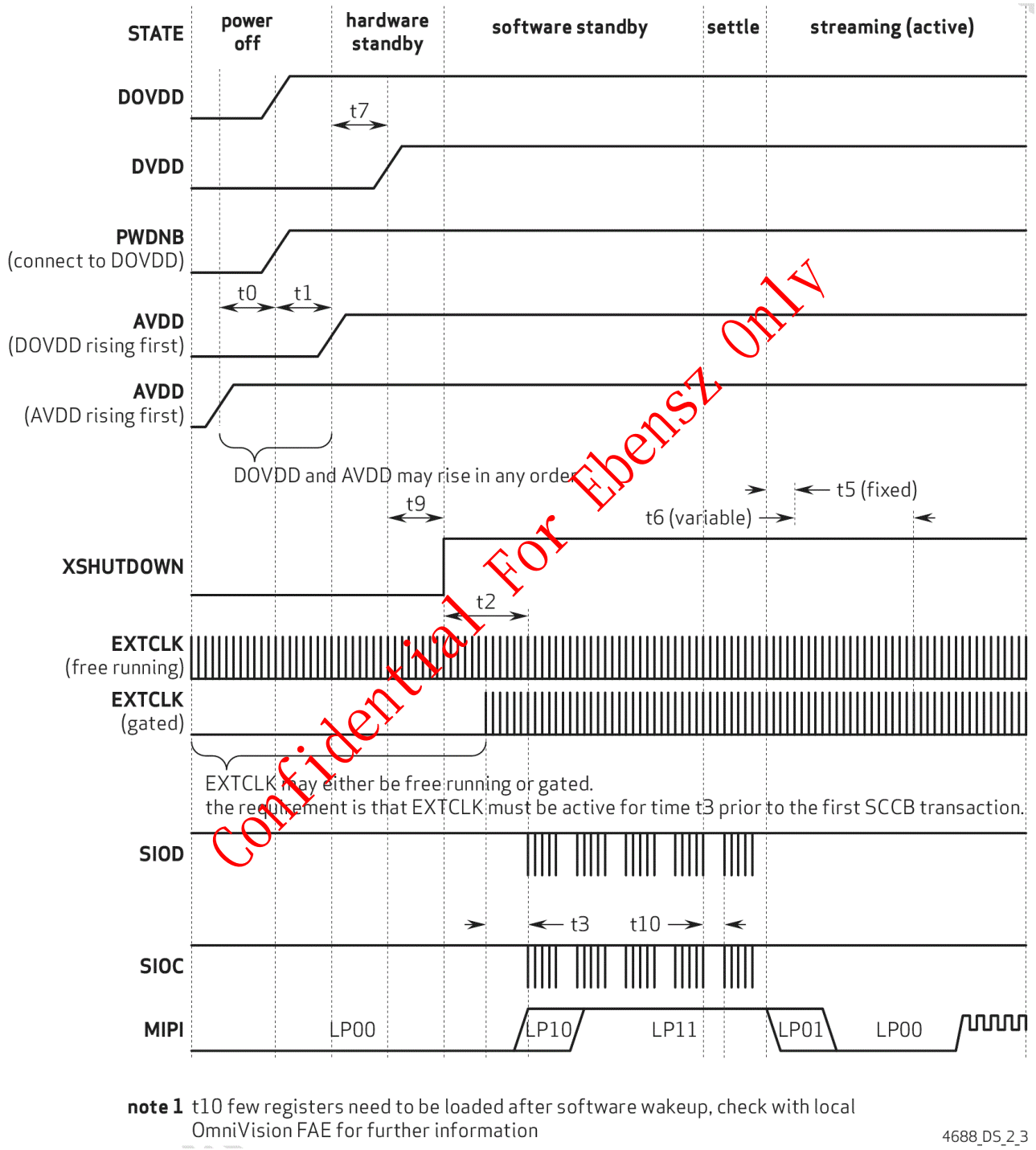
Confidential For Ebensz Only

1. Power up DOVDD and AVDD in any order or together(t0>=0, t1>=0). PWDNB is connected to DOVDD. Pull low XSHUTDOWN.

2. After t7, power up DVDD(t7>=0).

OV4688 is set into hardware standby mode. XVCLK could be free running or gated.

4.2.2 Enter Camera



Confidential For Ebensz Only

1. After t9 of DVDD power up, pull high XSHUTDOWN(t9>=0).
2. XVCLK could be free running or gated. If XVCLK is gated, apply XVCLK.

3. After 8192 XVCLK cycles(t3) of XVCLK applied and 8192 cycles (t2) of XSHUTDOWN pulled high, initialize OV4688 by SCCB writing. Please find OV4688 initialization setting from section 5.1.
4. After 10ms(t5) + exposure time(t6), the first frame image data outputs.

4.2.3 Switch to Other Camera

1. Software standby. Send SCCB registers to set OV4688 to software standby mode.
6c 0100 00
2. Delay 1 frame, then write following registers
6c 301a f9
6c 4805 03
if SCCB command received during the readout of the frame then the sensor must wait after the MIPI frame end short packet before entering sleep mode. If the SCCB command is received during the inter frame time, the sensor enter sleep mode immediately.

2. if XVCLK is gated, pull low XVCLK after 512 cycles of MIPI frame end

3. Wait for 1 frame time, enter other camera

4.2.4 Switch from Other Camera

1. power down other camera.

2. XVCLK could be free running or gated. If gated, provide XVCLK.

3. After 8192 cycles of XVCLK applied, write SCCB registers to wake up OV4688 from software power down.

6c 0100 01

6c 301a f9

4. After 10ms, write registers

6c 301a f1

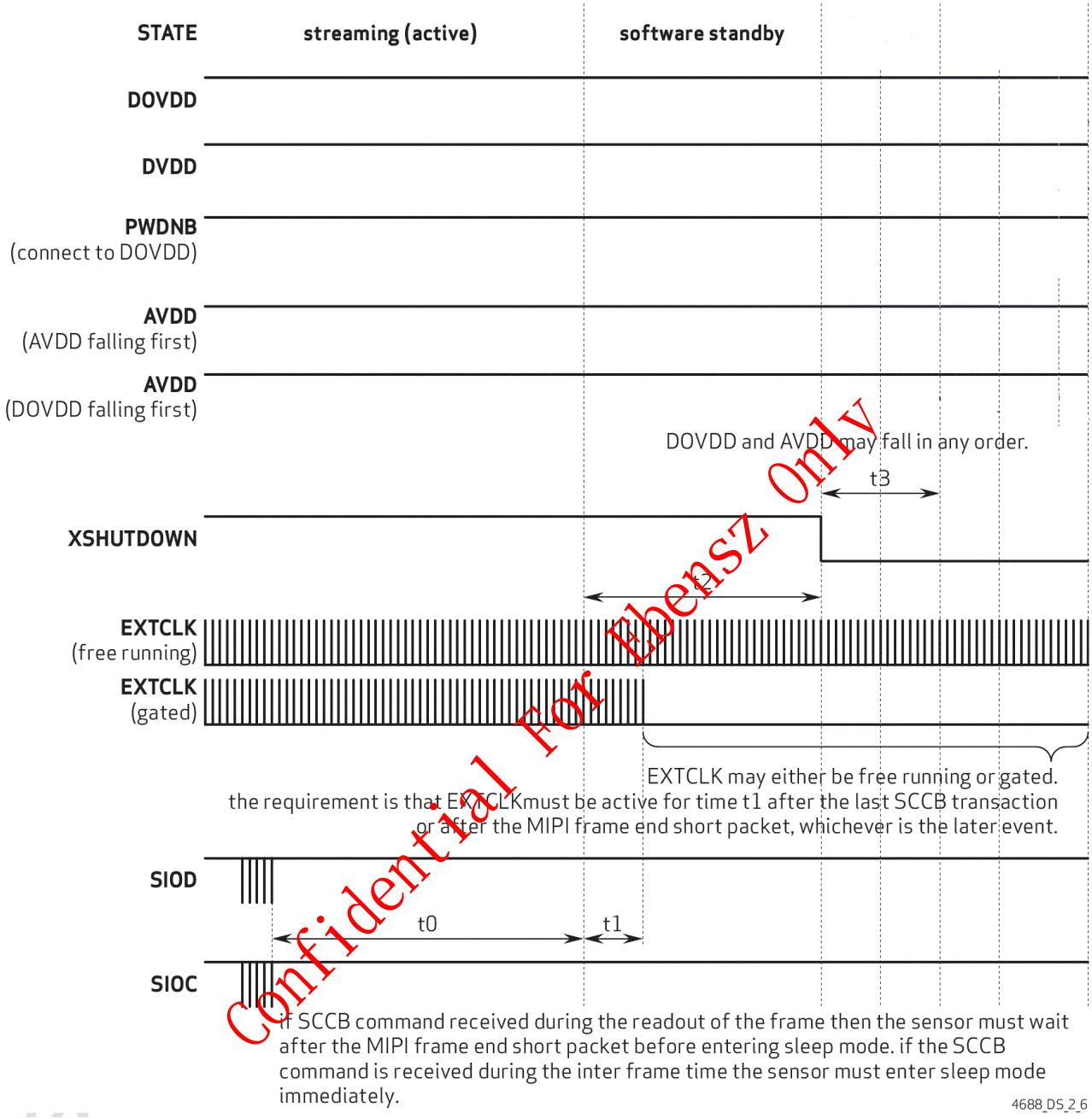
6c 4805 00

6c 301a f0

First frame image will output after 10ms(t5) + exposure time(t6).

Confidential For Ebensz Only

4.2.5 Exit Camera



Confidential For Ebensz Only

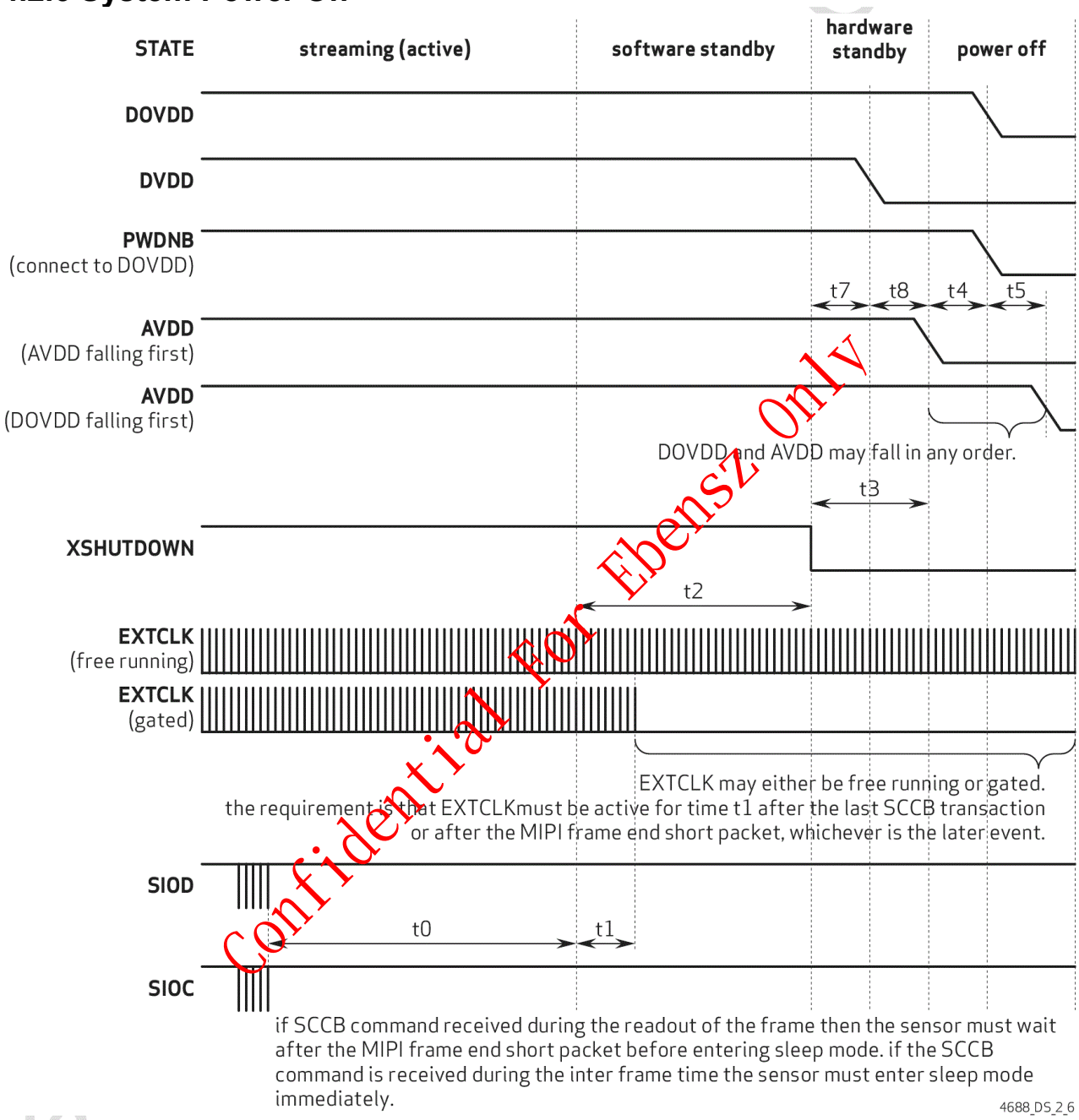
1. Software standby. Send SCCB registers to set OV4688 to software standby mode.
6c 0100 00
2. Delay 1 frame, then write following registers
6c 301a f9
6c 4805 03

if SCCB command received during the readout of the frame then the sensor must wait after the MIPI frame end short packet before entering sleep mode. If the SCCB command is received during the inter frame time, the sensor enter sleep mode immediately(t0).

2. if XVCLK is gated, pull low XVCLK after 512 cycles(t1) of MIPI frame end.

3. After 512 cycles(t2) of MIPI frame end, pull low XSHUTDOWN, set OV4688 to hardware power down mode.

4.2.6 System Power Off



Confidential For Ebensz Only

1. After t7(t7>=0) of XSHUTDOWN pulled low, turn off DVDD.
2. After t8(t8>=0), trun off DOVDD and AVDD. The 2 powers could be turned off in any order(t4>=0, t5>==0). PWDNB is connected to DOVDD.
2. If clock is gated, pull low XVCLK

When OV4688 is powered off, please pull low all input pins (except SCCB pins) to OV4688. If any input pin (except SCCB pins) of OV4688 is pulled high, the high input pin may leak current to DVDD of OV4688 and make the sensor unstable.

5. Software Operation of OV4688

Note:
1. The settings below are based on: “OV4688_AM19F.ovd”.
2. The following conditions are set for all settings
manual WB, manual AE, WBC on. BLC trig by Gain change and target value is 0x10.
3. The settings as organized as global settings and resolution settings. global setting is common to all resolutions and format, for specific resolution, users just need combining global setting and resolution setting. For example, if customer wants to get full size 30fps, 4 lane MIPI setting, user need to write “Initialization” followed by “Full Resolution 2688x1520 24fps 4 lane MIPI”.
4. Calibration data should be loaded as part of global setting.
5. The “initialization” setting reset all sensor registers at first. Then write related register settings. To change from one resolution to another, users doesn't need to reset all registers and write all register settings. Write resolution setting and format setting is enough for change resolution. For example, to change from 2688x1520 to 1080p 30fps video MIPI 4 lane, user just need to write “MIPI 1080p Full FOV 30fps 4 lane MIPI”.

5.1 Register Settings for 4 Lane MIPI, 1008Mbps

5.1.1 Initialization (Global Setting)

; MCLK = 24Mhz, Sysclk = 120Mhz, MIPI 4 lane, 1008 Mbps

```
42 0103 01      ; software reset
; delay 5ms
42 3638 00      ; ADC & analog
42 0300 00      ; PLL1 prediv
42 0302 2a      ; PLL1 divm
42 0304 03      ; PLL1 div mipi
42 030b 00      ; PLL2 pre div
42 030d 1e      ; PLL2 multiplier
42 030e 04      ; PLL2 divs
42 030f 01      ; PLL2 divsp
42 0312 01      ; PLL2 divdac
42 031e 00      ; Debug mode
42 3000 20      ; FSIN output
42 3002 00      ; Vsync input,  HREF input, FREX input, GPIO0 input
42 3018 72      ; MIPI 4 lane, Reset MIPI PHY when sleep
42 3020 93      ; Clock switch to PLL clock, Debug mode
42 3021 03      ; Sleep latch, software standby at line blank
42 3022 01      ; LVDS disable, Enable power down MIPI when sleep
42 3031 0a      ; MIPI 10-bit mode
42 3305 f1      ; ASRAM
```

Confidential For Ebensz Only

```
42 3307 04    ; ASRAM
42 3309 29    ; ASRAM
42 3500 00    ; Long exposure HH
42 3501 60    ; Long exposure H
42 3502 00    ; Long exposure L
42 3503 04    ; Gain delay 1 frame, use sensor gain, exposure delay 1 frame
42 3504 00    ; debug mode
42 3505 00    ; debug mode
42 3506 00    ; debug mode
42 3507 00    ; Long gain HH
42 3508 00    ; Long gain H
42 3509 80    ; Long gain L
42 350a 00    ; Middle exposure HH
42 350b 00    ; Middle exposure H
42 350c 00    ; Middle exposure L
42 350d 00    ; Middle gain HH
42 350e 00    ; Middle gain H
42 350f 80    ; Middle gain L
42 3510 00    ; Short exposure HH
42 3511 00    ; Short exposure H
42 3512 00    ; Short exposure L
42 3513 00    ; Short gain HH
42 3514 00    ; Short gain H
42 3515 80    ; Short gain L
42 3516 00    ; 4th exposure HH
42 3517 00    ; 4th exposure H
42 3518 00    ; 4th exposure L
42 3519 00    ; 4th gain HH
42 351a 00    ; 4th gain H
42 351b 80    ; 4th gian L
42 351c 00    ; 5th exposure HH
42 351d 00    ; 5th exposure H
42 351e 00    ; 5th exposure L
42 351f 00    ; 5th gain HH
```

Confidential For Ebensz Only

```
42 3520 00    ; 5th gain H
42 3521 80    ; 5th gain L
42 3522 08    ; Middle digital fraction gain H
42 3524 08    ; Short digital fraction gain H
42 3526 08    ; 4th digital fraction gain H
42 3528 08    ; 5th digital framction gain H
42 352a 08    ; Long digital fraction gain H
42 3602 00    ; ADC & analog
42 3604 02    ;
42 3605 00    ;
42 3606 00    ;
42 3607 00    ;
42 3609 12    ;
42 360a 40    ;
42 360c 08    ;
42 360f e5    ;
42 3608 8f    ;
42 3611 00    ;
42 3613 f7    ;
42 3616 58    ;
42 3619 99    ;
42 361b 60    ;
42 361c 7a    ;
42 361e 79    ;
42 361f 02    ;
42 3632 00    ;
42 3633 10    ;
42 3634 10    ;
42 3635 10    ;
42 3636 15    ;
42 3646 86    ;
42 364a 0b    ; ADC & analog
42 3700 17    ; Sensor control
42 3701 22    ;
```

Confidential For Ebensz Only

```
42 3703 10    ;
42 370a 37    ;
42 3705 00    ;
42 3706 63    ;
42 3709 3c    ;
42 370b 01    ;
42 370c 30    ;
42 3710 24    ;
42 3711 0c    ;
42 3716 00    ;
42 3720 28    ;
42 3729 7b    ;
42 372a 84    ;
42 372b bd    ;
42 372c bc    ;
42 372e 52    ;
42 373c 0e    ;
42 373e 33    ;
42 3743 10    ;
42 3744 88    ;
42 374a 43    ;
42 374c 00    ;
42 374e 23    ;
42 3751 7b    ;
42 3752 84    ;
42 3753 bd    ;
42 3754 bc    ;
42 3756 52    ;
42 375c 00    ;
42 3760 00    ;
42 3761 00    ;
42 3762 00    ;
42 3763 00    ;
42 3764 00    ;
```

Confidential For Ebensz Only

```
42 3767 04    ;
42 3768 04    ;
42 3769 08    ;
42 376a 08    ;
42 376b 20    ;
42 376c 00    ;
42 376d 00    ;
42 376e 00    ;
42 3773 00    ;
42 3774 51    ;
42 3776 bd    ;
42 3777 bd    ;
42 3781 18    ;
42 3783 25    ; Sensor control
42 3800 00    ; H crop start H
42 3801 08    ; H crop start L
42 3802 00    ; V crop start H
42 3803 04    ; V crop start L
42 3804 0a    ; H crop end H
42 3805 97    ; H crop end L
42 3806 05    ; V crop end H
42 3807 fb    ; V crop end L
42 3808 0a    ; H output size H
42 3809 80    ; H output size L
42 380a 05    ; V output size H
42 380b f0    ; V output size L
42 380c 03    ; HTS H
42 380d 5c    ; HTS L
42 380e 06    ; VTS H
42 380f 12    ; VTS L
42 3810 00    ; H win off H
42 3811 08    ; H win off L
42 3812 00    ; V win off H
42 3813 04    ; V win off L
```

Confidential For Ebensz Only

42 3814 01 ; H inc odd

42 3815 01 ; H inc even

42 3819 01 ; Vsync end L

42 3820 00 ; flip off, bin off

42 3821 06 ; mirror on, bin off

42 3829 00 ; HDR lite off

42 382a 01 ; vertical subsample odd increase number

42 382b 01 ; vertical subsample even increase number

42 382d 7f ; black column end address

42 3830 04 ; blc use num/2

42 3836 01 ; r zline use num/2

42 3841 02 ; r_rcnt_fix on

42 3846 08 ; fcnt_trig_rst_en on

42 3847 07 ; debug mode

42 3d85 36 ; OTP bist enable, OTP BIST compare with zero, OTP power up load data on, OTP power up load setting on, OTP write register load setting off

42 3d8c 71 ; OTP start address H

42 3d8d cb ; OTP start address L

42 3f0a 00 ; PSRAM

42 4000 71 ; out of range trig off, format chg on, gain chg on, exp chg on, manual trig off, no freeze, always trig off, debug mode

42 4001 40 ; debug mode

42 4002 04 ; debug mode

42 4003 14 ; black line number

42 400e 00 ; offset for BLC bypass

42 4011 00 ; offset man same off, offset man off, black line output off,

42 401a 00 ; debug mode

42 401b 00 ; debug mode

42 401c 00 ; debug mode

42 401d 00 ; debug mode

42 401f 00 ; debug mode

42 4020 00 ; Anchor left start H

42 4021 10 ; Anchor left start L

42 4022 07 ; Anchor left end H

Confidential For Ebensz Only

```
42 4023 cf      ; Anchor left end L
42 4024 09      ; Anchor right start H
42 4025 60      ; Andhor right start L
42 4026 09      ; Anchor right end H
42 4027 6f      ; Anchor right end L
42 4028 00      ; top Zline start
42 4029 02      ; top Zline number
42 402a 06      ; top blk line start
42 402b 04      ; to blk line number
42 402c 02      ; bot Zline start
42 402d 02      ; bot Zline number
42 402e 0e      ; bot blk line start
42 402f 04      ; bot blk line number
42 4302 ff      ; clipping max H
42 4303 ff      ; clipping max L
42 4304 00      ; clipping min H
42 4305 00      ; clipping min L
42 4306 00      ; vfifo pix swap off, dpcm off, vfifo first line is blue line
42 4308 02      ; debug mode, embeded off
42 4500 6c      ; ADC sync control
42 4501 c4      ;
42 4502 40      ;
42 4503 02      ; ADC sync control
42 4601 04      ; V fifo read start
42 4800 04      ; MIPI always high speed off, Clock lane gate off, line short packet off,
42 4813 08      ; Select HDR VC
42 481f 40      ; MIPI clock prepare min
42 4829 78      ; MIPI HS exit min
42 4837 10      ; MIPI global timing
42 4b00 2a
42 4b0d 00
42 4d00 04      ; tpm slope H
42 4d01 42      ; tpm slope L
42 4d02 d1      ; tpm offset HH
```

Confidential For Ebensz Only

```
42 4d03 93      ; tpm offset H
42 4d04 f5      ; tpm offset M
42 4d05 c1      ; tpm offset L
42 5000 f3      ; digital gain on, bin on, OTP on, WB gain on, average on, ISP on
42 5001 11      ; ISP EOF select, ISP SOF off, BLC on
42 5004 00      ; debug mode
42 500a 00      ; debug mode
42 500b 00      ; debug mode
42 5032 00      ; debug mode
42 5040 00      ; test mode off
42 5050 0c      ; debug mode
42 5500 00      ; OTP DPC start H
42 5501 10      ; OTP DPC start L
42 5502 01      ; OTP DPC end H
42 5503 0f      ; OTP DPC end L
42 8000 00      ; test mode
42 8001 00      ;
42 8002 00      ;
42 8003 00      ;
42 8004 00      ;
42 8005 00      ;
42 8006 00      ;
42 8007 00      ;
42 8008 00      ; test mode
42 3638 00      ; ADC & analog
42 3105 31      ; SCCB control, debug mode
42 301a f9      ; enable emb clock, enable strobe clock, enable timing control clock, mipi=phy
manual reset, reset timing control block
42 3508 07      ; Long gain H
42 484b 05      ; line start after fifo_st, sclock start after SOF, frame start after SOF
42 4805 03      ; MIPI control
42 3601 01      ; ADC & analog
42 0100 01      ; wake up from sleep
; Delay 10ms here
```

Confidential For Ebensz Only

```
42 3105 11      ; SCCB control, debug mode
42 301a f1      ; disable mipi-phy reset
42 4805 00      ; MIPI control
42 301a f0      ; enable emb clock, enable strobe clock, enable timing control clock,
42 3208 00      ; group hold start, group bank 0
42 302a 00      ; delay?
42 302a 00      ;
42 302a 00      ;
42 302a 00      ;
42 302a 00      ;
42 3601 00      ; ADC & analog
42 3638 00      ; ADC & analog
42 3208 10      ; group hold end, group select 0
42 3208 a0      ; group delay launch, group select 0
```

5.1.2 Full Resolution 2688x1520 90 fps

```
42 3501 60      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 00      ; H crop start H
42 3801 08      ; H crop start L
42 3803 04      ; V crop start L
42 3804 0a      ; H crop end H
42 3805 9f7     ; H crop end L
42 3807 fb      ; V crop end L
42 3808 0a      ; H output size H
42 3809 80      ; H output size L
42 380a 05      ; V output size H
42 380b f0      ; V output size L
42 380d 5c      ; HTS L
42 380e 06      ; VTS H
42 380f 12      ; VTS L
42 3811 08      ; H win off L
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
42 3820 00      ; flip off, bin off
42 3821 06      ; mirror on, bin off
42 382a 01      ; vertical subsample odd increase number
42 3830 04      ; blc use num/2
42 3836 01      ; r zline use num/2
42 4001 40      ; debug mode
42 4022 07      ; Anchor left end H
```

Confidential For Ebensz Only

```
42 4023 cf      ; Anchor left end L
42 4024 09      ; Anchor right start H
42 4025 60      ; Andhor right start L
42 4026 09      ; Anchor right end H
42 4027 6f      ; Anchor right end L
42 4502 40      ; ADC sync control
42 4601 04      ; V fifo read start
```

5.1.3 2016x1520 90 fps

```
42 3501 60      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 01      ; H crop start H
42 3801 58      ; H crop start L
42 3803 04      ; V crop start L
42 3804 09      ; H crop end H
42 3805 47      ; H crop end L
42 3807 fb      ; V crop end L
42 3808 07      ; H output size H
42 3809 e0      ; H output size L
42 380a 05      ; V output size H
42 380b f0      ; V output size L
42 380d 5c      ; HTS L
42 380e 06      ; VTS H
42 380f 12      ; VTS L
42 3811 08      ; H win off L
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
42 3820 00      ; flip off, bin off
42 3821 06      ; mirror on, bin off
42 382a 01      ; vertical subsample odd increase number
42 3830 04      ; blc use num/2
42 3836 01      ; r zline use num/2
42 4001 40      ; debug mode
42 4022 06      ; Anchor left end H
42 4023 3f      ; Anchor left end L
42 4024 07      ; Anchor right start H
42 4025 6c      ; Andhor right start L
42 4026 07      ; Anchor right end H
42 4027 7b      ; Anchor right end L
42 4502 40      ; ADC sync control
42 4601 04      ; V fifo read start
```

5.1.4 EIS 1080p 2120x1196 120fps

```
42 3501 4c      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
```

Confidential For Ebensz Only

```
42 3800 01    ; H crop start H
42 3801 24    ; H crop start L
42 3803 a6    ; V crop start L
42 3804 09    ; H crop end H
42 3805 7b    ; H crop end L
42 3807 59    ; V crop end L
42 3808 08    ; H output size H
42 3809 48    ; H output size L
42 380a 04    ; V output size H
42 380b ac    ; V output size L
42 380d 30    ; HTS L
42 380e 04    ; VTS H
42 380f ce    ; VTS L
42 3811 08    ; H win off L
42 3813 04    ; V win off L
42 3814 01    ; H inc odd
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 382a 01    ; vertical subsample odd increase number
42 3830 04    ; blc use num/2
42 3836 01    ; r zline use num/2
42 4001 40    ; debug mode
42 4022 06    ; Anchor left end H
42 4023 3f    ; Anchor left end L
42 4024 07    ; Anchor right start H
42 4025 6c    ; Andhor right start L
42 4026 07    ; Anchor right end H
42 4027 7b    ; Anchor right end L
42 4502 40    ; ADC sync control
42 4601 04    ; V fifo read start
```

5.1.5 EIS 720p 1344x760 180fps

```
42 3501 31    ; long exposure H
42 3632 05    ; ADC & Analog
42 376b 40    ; Sensor control
42 3800 00    ; H crop start H
42 3801 08    ; H crop start L
42 3803 04    ; V crop start L
42 3804 0a    ; H crop end H
42 3805 9f    ; H crop end L
42 3807 fb    ; V crop end L
42 3808 05    ; H output size H
42 3809 40    ; H output size L
42 380a 02    ; V output size H
42 380b f8    ; V output size L
42 380d 48    ; HTS L
42 380e 03    ; VTS H
42 380f 1d    ; VTS L
```

Confidential For Ebensz Only

```
42 3811 04    ; H win off L
42 3813 02    ; V win off L
42 3814 03    ; H inc odd
42 3820 10    ; flip off, bin on
42 3821 07    ; mirror on, bin on
42 382a 03    ; vertical subsample odd increase number
42 3830 08    ; blc use num/2
42 3836 02    ; r zline use num/2
42 4001 50    ; debug mode
42 4022 03    ; Anchor left end H
42 4023 e7    ; Anchor left end L
42 4024 05    ; Anchor right start H
42 4025 14    ; Andhor right start L
42 4026 05    ; Anchor right end H
42 4027 23    ; Anchor right end L
42 4502 44    ; ADC sync control
42 4601 28    ; V fifo read start
```

5.2 Register Setting for 4 Lane MIPI, 672Mbps

5.2.1 Initialization(Global Setting)

```
; MCLK=24Mhz, Sysclk = 120Mhz, MIPI 4 Lane, 672Mbps
42 0103 01    ; software reset
; delay 5ms
42 3638 00    ; ADC & analog
42 0300 00    ; PLL1 prediv
42 0302 1c    ; PLL1 divm
42 0304 03    ; PLL1 div mipi
42 030b 00    ; PLL2 pre div
42 030d 1e    ; PLL2 multiplier
42 030e 04    ; PLL2 divs
42 030f 01    ; PLL2 divsp
42 0312 01    ; PLL2 divdac
42 031e 00    ; Debug mode
42 3000 20    ; FSIN output
42 3002 00    ; Vsync input,  HREF input, FREX input, GPIO0 input
42 3018 72    ; MIPI 4 lane, Reset MIPI PHY when sleep
42 3020 93    ; Clock switch to PLL clock, Debug mode
42 3021 03    ; Sleep latch, software standby at line blank
42 3022 01    ; LVDS disable, Enable power down MIPI when sleep
42 3031 0a    ; MIPI 10-bit mode
42 3305 f1    ; ASRAM
42 3307 04    ; ASRAM
42 3309 29    ; ASRAM
42 3500 00    ; Long exposure HH
42 3501 60    ; Long exposure H
42 3502 00    ; Long exposure L
42 3503 04    ; Gain delay 1 frame, use sensor gain, exposure delay 1 frame
42 3504 00    ; debug mode
```

Confidential For Ebensz Only

```
42 3505 00    ; debug mode
42 3506 00    ; debug mode
42 3507 00    ; Long gain HH
42 3508 00    ; Long gain H
42 3509 80    ; Long gain L
42 350a 00    ; Middle exposure HH
42 350b 00    ; Middle exposure H
42 350c 00    ; Middle exposure L
42 350d 00    ; Middle gain HH
42 350e 00    ; Middle gain H
42 350f 80    ; Middle gain L
42 3510 00    ; Short exposure HH
42 3511 00    ; Short exposure H
42 3512 00    ; Short exposure L
42 3513 00    ; Short gain HH
42 3514 00    ; Short gain H
42 3515 80    ; Short gain L
42 3516 00    ; 4th exposure HH
42 3517 00    ; 4th exposure H
42 3518 00    ; 4th exposure L
42 3519 00    ; 4th gain HH
42 351a 00    ; 4th gain H
42 351b 80    ; 4th gian L
42 351c 00    ; 5th exposure HH
42 351d 00    ; 5th exposure H
42 351e 00    ; 5th exposure L
42 351f 00    ; 5th gain HH
42 3520 00    ; 5th gain H
42 3521 80    ; 5th gain L
42 3522 08    ; Middle digital fraction gain H
42 3524 08    ; Short digital fraction gain H
42 3526 08    ; 4th digital fraction gain H
42 3528 08    ; 5th digital framction gain H
42 352a 08    ; Long digital fraction gain H
42 3602 00    ; ADC & analog
42 3604 02    ;
42 3605 00    ;
42 3606 00    ;
42 3607 00    ;
42 3609 12    ;
42 360a 40    ;
42 360c 08    ;
42 360f e5    ;
42 3608 8f    ;
42 3611 00    ;
42 3613 f7    ;
42 3616 58    ;
42 3619 99    ;
```

Confidential For Ebensz Only

```
42 361b 60    ;
42 361c 7a    ;
42 361e 79    ;
42 361f 02    ;
42 3632 00    ;
42 3633 10    ;
42 3634 10    ;
42 3635 10    ;
42 3636 15    ;
42 3646 86    ;
42 364a 0b    ; ADC & analog
42 3700 17    ; Sensor control
42 3701 22    ;
42 3703 10    ;
42 370a 37    ;
42 3705 00    ;
42 3706 63    ;
42 3709 3c    ;
42 370b 01    ;
42 370c 30    ;
42 3710 24    ;
42 3711 0c    ;
42 3716 00    ;
42 3720 28    ;
42 3729 7b    ;
42 372a 84    ;
42 372b bd    ;
42 372c bc    ;
42 372e 52    ;
42 373c 0e    ;
42 373e 33    ;
42 3743 10    ;
42 3744 88    ;
42 374a 43    ;
42 374c 00    ;
42 374e 23    ;
42 3751 7b    ;
42 3752 84    ;
42 3753 bd    ;
42 3754 bc    ;
42 3756 52    ;
42 375c 00    ;
42 3760 00    ;
42 3761 00    ;
42 3762 00    ;
42 3763 00    ;
42 3764 00    ;
42 3767 04    ;
```

Confidential For Ebensz Only

```
42 3768 04    ;
42 3769 08    ;
42 376a 08    ;
42 376b 20    ;
42 376c 00    ;
42 376d 00    ;
42 376e 00    ;
42 3773 00    ;
42 3774 51    ;
42 3776 bd    ;
42 3777 bd    ;
42 3781 18    ;
42 3783 25    ; Sensor control
42 3800 00    ; H crop start H
42 3801 08    ; H crop start L
42 3802 00    ; V crop start H
42 3803 04    ; V crop start L
42 3804 0a    ; H crop end H
42 3805 97    ; H crop end L
42 3806 05    ; V crop end H
42 3807 fb    ; V crop end L
42 3808 0a    ; H output size H
42 3809 80    ; H output size L
42 380a 05    ; V output size H
42 380b f0    ; V output size L
42 380c 05    ; HTS H
42 380d 08    ; HTS L
42 380e 06    ; VTS H
42 380f 12    ; VTS L
42 3810 00    ; H win off H
42 3811 08    ; H win off L
42 3812 00    ; V win off H
42 3813 04    ; V win off L
42 3814 01    ; H inc odd
42 3815 01    ; H inc even
42 3819 01    ; Vsync end L
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 3829 00    ; HDR lite off
42 382a 01    ; vertical subsample odd increase number
42 382b 01    ; vertical subsample even increase number
42 382d 7f    ; black column end address
42 3830 04    ; blc use num/2
42 3836 01    ; r zline use num/2
42 3841 02    ; r_rcnt_fix on
42 3846 08    ; fcnt_trig_rst_en on
42 3847 07    ; debug mode
42 3d85 36    ; OTP bist enable, OTP BIST compare with zero, OTP power up load data on, OTP
```

Confidential For Ebensz Only

```
power up load setting on, OTP write register load setting off
42 3d8c 71      ; OTP start address H
42 3d8d cb      ; OTP start address L
42 3f0a 00      ; PSRAM
42 4000 71      ; out of range trig off, format chg on, gain chg on, exp chg on, manual trig off, no
freeze, always trig off, debug mode
42 4001 40      ; debug mode
42 4002 04      ; debug mode
42 4003 14      ; black line number
42 400e 00      ; offset for BLC bypass
42 4011 00      ; offset man same off, offset man off, black line output off,
42 401a 00      ; debug mode
42 401b 00      ; debug mode
42 401c 00      ; debug mode
42 401d 00      ; debug mode
42 401f 00      ; debug mode
42 4020 00      ; Anchor left start H
42 4021 10      ; Anchor left start L
42 4022 07      ; Anchor left end H
42 4023 cf      ; Anchor left end L
42 4024 09      ; Anchor right start H
42 4025 60      ; Andhor right start L
42 4026 09      ; Anchor right end H
42 4027 6f      ; Anchor right end L
42 4028 00      ; top Zline start
42 4029 02      ; top Zline number
42 402a 06      ; top blk line start
42 402b 04      ; to blk line number
42 402c 02      ; bot Zline start
42 402d 02      ; bot Zline number
42 402e 0e      ; bot blk line start
42 402f 04      ; bot blk line number
42 4302 ff      ; clipping max H
42 4303 ff      ; clipping max L
42 4304 00      ; clipping min H
42 4305 00      ; clipping min L
42 4306 00      ; vfifo pix swap off, dpcm off, vfifo first line is blue line
42 4308 02      ; debug mode, embeded off
42 4500 6c      ; ADC sync control
42 4501 c4      ;
42 4502 40      ;
42 4503 02      ; ADC sync control
42 4601 04      ; V fifo read start
42 4800 04      ; MIPI always high speed off, Clock lane gate off, line short packet off,
42 4813 08      ; Select HDR VC
42 481f 40      ; MIPI clock prepare min
42 4829 78      ; MIPI HS exit min
42 4837 18      ; MIPI global timing
```

Confidential For Ebensz Only

```
42 4b00 2a
42 4b0d 00
42 4d00 04      ; tpm slope H
42 4d01 42      ; tpm slope L
42 4d02 d1      ; tpm offset HH
42 4d03 93      ; tpm offset H
42 4d04 f5      ; tpm offset M
42 4d05 c1      ; tpm offset L
42 5000 f3      ; digital gain on, bin on, OTP on, WB gain on, average on, ISP on
42 5001 11      ; ISP EOF select, ISP SOF off, BLC on
42 5004 00      ; debug mode
42 500a 00      ; debug mode
42 500b 00      ; debug mode
42 5032 00      ; debug mode
42 5040 00      ; test mode off
42 5050 0c      ; debug mode
42 5500 00      ; OTP DPC start H
42 5501 10      ; OTP DPC start L
42 5502 01      ; OTP DPC end H
42 5503 0f      ; OTP DPC end L
42 8000 00      ; test mode
42 8001 00      ;
42 8002 00      ;
42 8003 00      ;
42 8004 00      ;
42 8005 00      ;
42 8006 00      ;
42 8007 00      ;
42 8008 00      ; test mode
42 3638 00      ; ADC & analog
42 3105 31      ; SCCB control, debug mode
42 301a f9      ; enable emb clock, enable strobe clock, enable timing control clock, mipi-phy
manual reset, reset timing control block
42 3508 07      ; Long gain H
42 484b 05      ; line start after fifo_st, sclock start after SOF, frame start after SOF
42 4805 03      ; MIPI control
42 3601 01      ; ADC & analog
42 0100 01      ; wake up from sleep
; delay 2ms here
42 3105 11      ; SCCB control, debug mode
42 301a f1      ; disable mipi-phy reset
42 4805 00      ; MIPI control
42 301a f0      ; enable emb clock, enable strobe clock, enable timing control clock,
42 3208 00      ; group hold start, group bank 0
42 302a 00      ; delay?
42 302a 00      ;
42 302a 00      ;
42 302a 00      ;
```

Confidential For Ebensz Only

```
42 302a 00      ;
42 3601 00      ; ADC & analog
42 3638 00      ; ADC & analog
42 3208 10      ; group hold end, group select 0
42 3208 a0      ; group delay launch, group select 0
```

5.2.1 Full Resolution 2688x1520 60fps

```
42 3501 60      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 00      ; H crop start H
42 3801 08      ; H crop start L
42 3803 04      ; V crop start L
42 3804 0a      ; H crop end H
42 3805 97      ; H crop end L
42 3807 fb      ; V crop end L
42 3808 0a      ; H output size H
42 3809 80      ; H output size L
42 380a 05      ; V output size H
42 380b f0      ; V output size L
42 380d 08      ; HTS L
42 380e 06      ; VTS H
42 380f 12      ; VTS L
42 3811 08      ; H win off L
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
42 3820 00      ; flip off, bin off
42 3821 06      ; mirror on, bin off
42 382a 01      ; vertical subsample odd increase number
42 3830 04      ; blc use num/2
42 3836 01      ; r zline use num/2
42 4001 40      ; debug mode
42 4022 07      ; Anchor left end H
42 4023 cf      ; Anchor left end L
42 4024 09      ; Anchor right start H
42 4025 60      ; Andhor right start L
42 4026 09      ; Anchor right end H
42 4027 6f      ; Anchor right end L
42 4502 40      ; ADC sync control
42 4601 04      ; V fifo read start
```

5.2.2 2016x1520 60fps

```
42 3501 60      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 01      ; H crop start H
42 3801 58      ; H crop start L
42 3803 04      ; V crop start L
```

Confidential For Ebensz Only

```
42 3804 09      ; H crop end H
42 3805 47      ; H crop end L
42 3807 fb      ; V crop end L
42 3808 07      ; H output size H
42 3809 e0      ; H output size L
42 380a 05      ; V output size H
42 380b f0      ; V output size L
42 380d 08      ; HTS L
42 380e 06      ; VTS H
42 380f 12      ; VTS L
42 3811 08      ; H win off L
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
42 3820 00      ; flip off, bin off
42 3821 06      ; mirror on, bin off
42 382a 01      ; vertical subsample odd increase number
42 3830 04      ; blc use num/2
42 3836 01      ; r zline use num/2
42 4001 40      ; debug mode
42 4022 06      ; Anchor left end H
42 4023 3f      ; Anchor left end L
42 4024 07      ; Anchor right start H
42 4025 6c      ; Andhor right start L
42 4026 07      ; Anchor right end H
42 4027 7b      ; Anchor right end L
42 4502 40      ; ADC sync control
42 4601 04      ; V fifo read start
```

5.2.3 EIS 1080p 2120x1196 75fps

```
42 3501 4c      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 01      ; H crop start H
42 3801 24      ; H crop start L
42 3803 a6      ; V crop start L
42 3804 09      ; H crop end H
42 3805 7b      ; H crop end L
42 3807 59      ; V crop end L
42 3808 08      ; H output size H
42 3809 48      ; H output size L
42 380a 04      ; V output size H
42 380b ac      ; V output size L
42 380d 14      ; HTS L
42 380e 04      ; VTS H
42 380f ce      ; VTS L
42 3811 08      ; H win off L
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
```

Confidential For Ebensz Only

```
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 382a 01    ; vertical subsample odd increase number
42 3830 04    ; blc use num/2
42 3836 01    ; r zline use num/2
42 4001 40    ; debug mode
42 4022 06    ; Anchor left end H
42 4023 3f    ; Anchor left end L
42 4024 07    ; Anchor right start H
42 4025 6c    ; Andhor right start L
42 4026 07    ; Anchor right end H
42 4027 7b    ; Anchor right end L
42 4502 40    ; ADC sync control
42 4601 04    ; V fifo read start
```

5.2.4 EIS 720p 1344x760 110fps

```
42 3501 31    ; long exposure H
42 3632 05    ; ADC & Analog
42 376b 40    ; Sensor control
42 3800 00    ; H crop start H
42 3801 08    ; H crop start L
42 3803 04    ; V crop start L
42 3804 0a    ; H crop end H
42 3805 9f    ; H crop end L
42 3807 fb    ; V crop end L
42 3808 05    ; H output size H
42 3809 40    ; H output size L
42 380a 02    ; V output size H
42 380b f8    ; V output size L
42 380d 58    ; HTS L
42 380e 03    ; VTS H
42 380f 1d    ; VTS L
42 3811 08    ; H win off L
42 3813 02    ; V win off L
42 3814 03    ; H inc odd
42 3820 10    ; flip off, bin on
42 3821 07    ; mirror on, bin on
42 382a 03    ; vertical subsample odd increase number
42 3830 08    ; blc use num/2
42 3836 02    ; r zline use num/2
42 4001 50    ; debug mode
42 4022 03    ; Anchor left end H
42 4023 e7    ; Anchor left end L
42 4024 05    ; Anchor right start H
42 4025 14    ; Andhor right start L
42 4026 05    ; Anchor right end H
42 4027 23    ; Anchor right end L
42 4502 44    ; ADC sync control
```

Confidential For Ebensz Only

```
42 4601 28      ; V fifo read start
```

5.3 Settings for 2 Lane MIPI, 672Mbps

5.3.1 Initialization (Global Settings)

```
42 0103 01      ; software reset
42 3638 00      ; ADC & analog
42 0300 00      ; PLL1 prediv
42 0302 1c      ; PLL1 divm
42 0304 03      ; PLL1 div mipi
42 030b 00      ; PLL2 pre div
42 030d 1e      ; PLL2 multiplier
42 030e 04      ; PLL2 divs
42 030f 01      ; PLL2 divsp
42 0312 01      ; PLL2 divdac
42 031e 00      ; Debug mode
42 3000 20      ; FSIN output
42 3002 00      ; Vsync input,  HREF input, FREX input, GPIO0 input
42 3018 32      ; MIPI 2 lane, Reset MIPI PHY when sleep
42 3020 93      ; Clock switch to PLL clock, Debug mode
42 3021 03      ; Sleep latch, software standby at line blank
42 3022 01      ; LVDS disable, Enable power down MIPI when sleep
42 3031 0a      ; MIPI 10-bit mode
42 3305 f1      ; ASRAM
42 3307 04      ; ASRAM
42 3309 29      ; ASRAM
42 3500 00      ; Long exposure HH
42 3501 60      ; Long exposure H
42 3502 00      ; Long exposure L
42 3503 04      ; Gain delay 1 frame, use sensor gain, exposure delay 1 frame
42 3504 00      ; debug mode
42 3505 00      ; debug mode
42 3506 00      ; debug mode
42 3507 00      ; Long gain HH
42 3508 00      ; Long gain H
42 3509 80      ; Long gain L
42 350a 00      ; Middle exposure HH
42 350b 00      ; Middle exposure H
42 350c 00      ; Middle exposure L
42 350d 00      ; Middle gain HH
42 350e 00      ; Middle gain H
42 350f 80      ; Middle gain L
42 3510 00      ; Short exposure HH
42 3511 00      ; Short exposure H
42 3512 00      ; Short exposure L
42 3513 00      ; Short gain HH
42 3514 00      ; Short gain H
42 3515 80      ; Short gain L
42 3516 00      ; 4th exposure HH
```

Confidential For Ebensz Only

```
42 3517 00    ; 4th exposure H
42 3518 00    ; 4th exposure L
42 3519 00    ; 4th gain HH
42 351a 00    ; 4th gain H
42 351b 80    ; 4th gian L
42 351c 00    ; 5th exposure HH
42 351d 00    ; 5th exposure H
42 351e 00    ; 5th exposure L
42 351f 00    ; 5th gain HH
42 3520 00    ; 5th gain H
42 3521 80    ; 5th gain L
42 3522 08    ; Middle digital fraction gain H
42 3524 08    ; Short digital fraction gain H
42 3526 08    ; 4th digital fraction gain H
42 3528 08    ; 5th digital framction gain H
42 352a 08    ; Long digital fraction gain H
42 3602 00    ; ADC & analog
42 3604 02    ;
42 3605 00    ;
42 3606 00    ;
42 3607 00    ;
42 3609 12    ;
42 360a 40    ;
42 360c 08    ;
42 360f e5    ;
42 3608 8f    ;
42 3611 00    ;
42 3613 f7    ;
42 3616 58    ;
42 3619 99    ;
42 361b 60    ;
42 361c 7a    ;
42 361e 79    ;
42 361f 02    ;
42 3632 00    ;
42 3633 10    ;
42 3634 10    ;
42 3635 10    ;
42 3636 15    ;
42 3646 86    ;
42 364a 0b    ; ADC & analog
42 3700 17    ; Sensor control
42 3701 22    ;
42 3703 10    ;
42 370a 37    ;
42 3705 00    ;
42 3706 63    ;
42 3709 3c    ;
```

Confidential For Ebensz Only

```
42 370b 01    ;
42 370c 30    ;
42 3710 24    ;
42 3711 0c    ;
42 3716 00    ;
42 3720 28    ;
42 3729 7b    ;
42 372a 84    ;
42 372b bd    ;
42 372c bc    ;
42 372e 52    ;
42 373c 0e    ;
42 373e 33    ;
42 3743 10    ;
42 3744 88    ;
42 374a 43    ;
42 374c 00    ;
42 374e 23    ;
42 3751 7b    ;
42 3752 84    ;
42 3753 bd    ;
42 3754 bc    ;
42 3756 52    ;
42 375c 00    ;
42 3760 00    ;
42 3761 00    ;
42 3762 00    ;
42 3763 00    ;
42 3764 00    ;
42 3767 04    ;
42 3768 04    ;
42 3769 08    ;
42 376a 08    ;
42 376b 20    ;
42 376c 00    ;
42 376d 00    ;
42 376e 00    ;
42 3773 00    ;
42 3774 51    ;
42 3776 bd    ;
42 3777 bd    ;
42 3781 18    ;
42 3783 25    ; Sensor control
42 3800 00    ; H crop start H
42 3801 08    ; H crop start L
42 3802 00    ; V crop start H
42 3803 04    ; V crop start L
42 3804 0a    ; H crop end H
```

Confidential For Ebensz Only

```
42 3805 97      ; H crop end L
42 3806 05      ; V crop end H
42 3807 fb      ; V crop end L
42 3808 0a      ; H output size H
42 3809 80      ; H output size L
42 380a 05      ; V output size H
42 380b f0      ; V output size L
42 380c 0a      ; HTS H
42 380d 10      ; HTS L
42 380e 06      ; VTS H
42 380f 12      ; VTS L
42 3810 00      ; H win off H
42 3811 08      ; H win off L
42 3812 00      ; V win off H
42 3813 04      ; V win off L
42 3814 01      ; H inc odd
42 3815 01      ; H inc even
42 3819 01      ; Vsync end L
42 3820 00      ; flip off, bin off
42 3821 06      ; mirror on, bin off
42 3829 00      ; HDR lite off
42 382a 01      ; vertical subsample odd increase number
42 382b 01      ; vertical subsample even increase number
42 382d 7f      ; black column end address
42 3830 04      ; blc use num/2
42 3836 01      ; r zline use num/2
42 3841 02      ; r_rcnt_fix on
42 3846 08      ; fcnt_trig_rst_en on
42 3847 07      ; debug mode
42 3d85 36      ; OTP bist enable, OTP BIST compare with zero, OTP power up load data on, OTP
power up load setting on, OTP write register load setting off
42 3d8c 71      ; OTP start address H
42 3d8d cb      ; OTP start address L
42 3f0a 00      ; PSRAM
42 4000 71      ; out of range trig off, format chg on, gain chg on, exp chg on, manual trig off, no
freeze, always trig off, debug mode
42 4001 40      ; debug mode
42 4002 04      ; debug mode
42 4003 14      ; black line number
42 400e 00      ; offset for BLC bypass
42 4011 00      ; offset man same off, offset man off, black line output off,
42 401a 00      ; debug mode
42 401b 00      ; debug mode
42 401c 00      ; debug mode
42 401d 00      ; debug mode
42 401f 00      ; debug mode
42 4020 00      ; Anchor left start H
42 4021 10      ; Anchor left start L
```

Confidential For Ebensz Only

```
42 4022 07    ; Anchor left end H
42 4023 cf    ; Anchor left end L
42 4024 09    ; Anchor right start H
42 4025 60    ; Andhor right start L
42 4026 09    ; Anchor right end H
42 4027 6f    ; Anchor right end L
42 4028 00    ; top Zline start
42 4029 02    ; top Zline number
42 402a 06    ; top blk line start
42 402b 04    ; to blk line number
42 402c 02    ; bot Zline start
42 402d 02    ; bot Zline number
42 402e 0e    ; bot blk line start
42 402f 04    ; bot blk line number
42 4302 ff    ; clipping max H
42 4303 ff    ; clipping max L
42 4304 00    ; clipping min H
42 4305 00    ; clipping min L
42 4306 00    ; vfifo pix swap off, dpcm off, vfifo first line is blue line
42 4308 02    ; debug mode, embeded off
42 4500 6c    ; ADC sync control
42 4501 c4    ;
42 4502 40    ;
42 4503 02    ; ADC sync control
42 4601 04    ; V fifo read start
42 4800 04    ; MIPI always high speed off, Clock lane gate off, line short packet off,
42 4813 08    ; Select HDR VC
42 481f 40    ; MIPI clock prepare min
42 4829 78    ; MIPI HS exit min
42 4837 18    ; MIPI global timing
42 4b00 2a
42 4b0d 00
42 4d00 04    ; tpm slope H
42 4d01 42    ; tpm slope L
42 4d02 d1    ; tpm offset HH
42 4d03 93    ; tpm offset H
42 4d04 f5    ; tpm offset M
42 4d05 c1    ; tpm offset L
42 5000 f3    ; digital gain on, bin on, OTP on, WB gain on, average on, ISP on
42 5001 11    ; ISP EOF select, ISP SOF off, BLC on
42 5004 00    ; debug mode
42 500a 00    ; debug mode
42 500b 00    ; debug mode
42 5032 00    ; debug mode
42 5040 00    ; test mode off
42 5050 0c    ; debug mode
42 5500 00    ; OTP DPC start H
42 5501 10    ; OTP DPC start L
```

Confidential For Ebensz Only

```
42 5502 01      ; OTP DPC end H
42 5503 0f      ; OTP DPC end L
42 8000 00      ; test mode
42 8001 00      ;
42 8002 00      ;
42 8003 00      ;
42 8004 00      ;
42 8005 00      ;
42 8006 00      ;
42 8007 00      ;
42 8008 00      ; test mode
42 3638 00      ; ADC & analog
42 3105 31      ; SCCB control, debug mode
42 301a f9      ; enable emb clock, enable strobe clock, enable timing control clock, mipi-phy
manual reset, reset timing control block
42 3508 07      ; Long gain H
42 484b 05      ; line start after fifo_st, sclock start after SOF, frame start after SOF
42 4805 03      ; MIPI control
42 3601 01      ; ADC & analog
42 0100 01      ; wake up from sleep
; delay 2ms here
42 3105 11      ; SCCB control, debug mode
42 301a f1      ; disable mipi-phy reset
42 4805 00      ; MIPI control
42 301a f0      ; enable emb clock, enable strobe clock, disable mipi-phy reset, enable timing control
clock,
42 3208 00      ; group hold start, group bank 0
42 302a 00      ; delay?
42 302a 00      ;
42 302a 00      ;
42 302a 00      ;
42 302a 00      ;
42 3601 00      ; ADC & analog
42 3638 00      ; ADC & analog
42 3208 10      ; group hold end, group select 0
42 3208 a0      ; group delay launch, group select 0
```

5.3.2 Full Resolution 2688x1520 30fps

```
42 3501 60      ; long exposure H
42 3632 00      ; ADC & Analog
42 376b 20      ; Sensor control
42 3800 00      ; H crop start H
42 3801 08      ; H crop start L
42 3803 04      ; V crop start L
42 3804 0a      ; H crop end H
42 3805 97      ; H crop end L
42 3807 fb      ; V crop end L
42 3808 0a      ; H output size H
```

Confidential For Ebensz Only

```
42 3809 80    ; H output size L
42 380a 05    ; V output size H
42 380b f0    ; V output size L
42 380d 10    ; HTS L
42 380e 06    ; VTS H
42 380f 12    ; VTS L
42 3811 08    ; H win off L
42 3813 04    ; V win off L
42 3814 01    ; H inc odd
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 382a 01    ; vertical subsample odd increase number
42 3830 04    ; blc use num/2
42 3836 01    ; r zline use num/2
42 4001 40    ; debug mode
42 4022 07    ; Anchor left end H
42 4023 cf    ; Anchor left end L
42 4024 09    ; Anchor right start H
42 4025 60    ; Andhor right start L
42 4026 09    ; Anchor right end H
42 4027 6f    ; Anchor right end L
42 4502 40    ; ADC sync control
42 4601 04    ; V fifo read start
```

5.3.3 2016x1520 30fps

```
42 3501 60    ; long exposure H
42 3632 00    ; ADC & Analog
42 376b 20    ; Sensor control
42 3800 01    ; H crop start H
42 3801 58    ; H crop start L
42 3803 04    ; V crop start L
42 3804 09    ; H crop end H
42 3805 47    ; H crop end L
42 3807 fb    ; V crop end L
42 3808 07    ; H output size H
42 3809 e0    ; H output size L
42 380a 05    ; V output size H
42 380b f0    ; V output size L
42 380d 10    ; HTS L
42 380e 06    ; VTS H
42 380f 12    ; VTS L
42 3811 08    ; H win off L
42 3813 04    ; V win off L
42 3814 01    ; H inc odd
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 382a 01    ; vertical subsample odd increase number
42 3830 04    ; blc use num/2
```

Confidential For Ebensz Only

```
42 3836 01    ; r zline use num/2
42 4001 40    ; debug mode
42 4022 06    ; Anchor left end H
42 4023 3f    ; Anchor left end L
42 4024 07    ; Anchor right start H
42 4025 6c    ; Andhor right start L
42 4026 07    ; Anchor right end H
42 4027 7b    ; Anchor right end L
42 4502 40    ; ADC sync control
42 4601 04    ; V fifo read start
```

5.3.4 EIS 1080p 2120x1196 30fps

```
42 3501 4c    ; long exposure H
42 3632 00    ; ADC & Analog
42 376b 20    ; Sensor control
42 3800 01    ; H crop start H
42 3801 24    ; H crop start L
42 3803 a6    ; V crop start L
42 3804 09    ; H crop end H
42 3805 7b    ; H crop end L
42 3807 59    ; V crop end L
42 3808 08    ; H output size H
42 3809 48    ; H output size L
42 380a 04    ; V output size H
42 380b ac    ; V output size L
42 380d 30    ; HTS L
42 380e 04    ; VTS H
42 380f ce    ; VTS L
42 3811 08    ; H win off L
42 3813 04    ; V win off L
42 3814 01    ; H inc odd
42 3820 00    ; flip off, bin off
42 3821 06    ; mirror on, bin off
42 382a 01    ; vertical subsample odd increase number
42 3830 04    ; blc use num/2
42 3836 01    ; r zline use num/2
42 4001 40    ; debug mode
42 4022 06    ; Anchor left end H
42 4023 3f    ; Anchor left end L
42 4024 07    ; Anchor right start H
42 4025 6c    ; Andhor right start L
42 4026 07    ; Anchor right end H
42 4027 7b    ; Anchor right end L
42 4502 40    ; ADC sync control
42 4601 04    ; V fifo read start
```

5.3.5 EIS 720p 1344x760 58.5fps

```
42 3501 31    ; long exposure H
```

Confidential For Ebensz Only

```
42 3632 05      ; ADC & Analog
42 376b 40      ; Sensor control
42 3800 00      ; H crop start H
42 3801 08      ; H crop start L
42 3803 04      ; V crop start L
42 3804 0a      ; H crop end H
42 3805 9f      ; H crop end L
42 3807 fb      ; V crop end L
42 3808 05      ; H output size H
42 3809 40      ; H output size L
42 380a 02      ; V output size H
42 380b f8      ; V output size L
42 380d 48      ; HTS L
42 380e 03      ; VTS H
42 380f 1d      ; VTS L
42 3811 04      ; H win off L
42 3813 02      ; V win off L
42 3814 03      ; H inc odd
42 3820 10      ; flip off, bin on
42 3821 07      ; mirror on, bin on
42 382a 03      ; vertical subsample odd increase number
42 3830 08      ; blc use num/2
42 3836 02      ; r zline use num/2
42 4001 50      ; debug mode
42 4022 03      ; Anchor left end H
42 4023 e7      ; Anchor left end L
42 4024 05      ; Anchor right start H
42 4025 14      ; Andhor right start L
42 4026 05      ; Anchor right end H
42 4027 23      ; Anchor right end L
42 4502 44      ; ADC sync control
42 4601 28      ; V fifo read start
```

Confidential For Ebensz Only

5.4 Drive capability

Table – 7 Drive capability

Register bit	function	Description
0x3011[6:5]	Pad I/O Drive capability	00: 1x 01: 2x 10: 3x 11: 4x

5.5 I/O Control

Table – 8 I/O Control

Signal	Input/High-z	Output data path	Output-0	Output-1
VSYNC	3002[7]=0	3002[7]=1	3002[7]=1	3002[7]=1

		3010[7]=0	3010[7]=1 300d[7]=0	3010[7]=1 300d[7]=1
HREF	3002[6]=0	3002[6]=1 3010[6]=0	3002[6]=1 3010[6]=1 300d[6]=0	3002[6]=1 3010[6]=1 300d[6]=1
FREX	3002[4]=0	3002[4]=1 3010[4]=0	3002[4]=1 3010[4]=1 300d[4]=0	3002[4]=1 3010[4]=1 300d[4]=1
GPIO0	3002[0]=0		3002[0]=1 3010[0]=1 300d[0]=0	3002[0]=1 3010[0]=1 300d[0]=1

5.6 MIPI Stream Control

MIPI stream could be controlled either by register 0x4202 or register 0x0100.

Controlled by register 0x4202

Stream On

20 4202 00 ; MIPI stream on when new frame starts

Stream Off

20 4202 0f ; MIPI stream off when current frame finish. Both *clock* and data lane in LP11 mode.

Controlled by register 0x0100

Stream On

20 0100 01 ; Sensor wake up from standby, MIPI stream on

Stream Off

20 0100 00 ; Sensor go to standby. MIPI stream off, both *clock* and data lane in LP11 mode.

5.7 Mirror and Flip

Note:

OV4688 has BLC performance issue when flip is turned on. If flip is turned on, need backend ISP to support:

1. Different BLC offset compensatioon between preview(binning mode) and capture(full resolution) mode.

2. Different BLC offset for differrent color channel.

Confidential For Ebensz Only

Full Size 4M/3M

```
#ifdef FLIP
        #ifdef MIRROR
                42  3820  06
                42  3821  06
        #else
                42  3820  06
                42  3821  00
        #endif
#else
        #ifdef MIRROR
                42  3820  00
                42  3821  06
        #else
                42  3820  00
                24  3821  00
        #endif
#endif
```

Quarter Size with H/V Binnign 720p or smaller

```
#ifdef FLIP
        #ifdef MIRROR
                42  3820  16
                42  3821  07
        #else
                42  3820  16
                42  3821  01
        #endif
#else
        #ifdef MIRROR
                42  3820  10
                42  3821  07
        #else
                42  3820  10
                42  3821  01
        #endif
#endif
```

Confidential For Ebensz Only

5.8 BLC (Black Level Calibration) Strategy

The function of Black Level Calibration (BLC) is match electric black of OV4688 output to optical black of sensor array to get accurate color in the dark area of picture. OV4688 has built-in automatic BLC function. It should always be turned on.

```
42 4000 71      ; offset out of range trigger off
                ; format change trigger on
                ; gain change trigger on
                ; exposure change trigger on
                ; manually trigger off
```

```
                ; BLC freeze off
                ; BLC always trigger off
42 4004 00      ; BLC target H
42 4005 40      ; BLC target L
```

Note;
OV4688 has bug in BLC offset out of tange trigger. To compensate BLC shift caused by junction temperature change, following solutions are recommended.
Solution 1:
Use backend ISP periodically do manual BLC trigger every 1~2 seconds.
Solution 2:
Monitor sensor's junction temperature by reading register 0x4d13(nned to set register 0x4d12 to be "1" to trigger junction temperature read out first). If the change is ~2 Deg C, manually trigger BLC once.

Confidential For Ebensz Only

5.9 Load Calibration Data

5.2.1 Calibration Data Assignment

The calibration data are stored in OTP of OV4688 by 3 segments. Module integrator ID, Lens ID, and manufacture data are stored in segment 1. AWB calibration data is stored in segment 2. VCM calibration code is stored in segment 3. Each segment support 3 data groups so that 2 times of re-write are supported for camera module rework.

Segment 1: Base Information

Flag	1 Byte
Module integrator ID	1 Byte
Lens ID	1 Byte
Module Production Year	1 Byte
Module Production Month	1 Byte
Module Production Day	1 Byte
AWB ratio R/G MSB	1 Byte
AWB ratio B/G MSB	1 Byte
Light source ratio R/G MSB	1 Byte
Light source ratio B/G MSB	1 Byte
AWB ratio R/G LSB	2 bit
AWB ratio B/G LSB	2 bit
Light source ratio R/G LSB	2 bit
Light source ratio B/G LSB	2 bit
User Data 1	1 Byte
User Data 2	1 Byte
User Data 3	1 Byte
User Data 4	1 Byte
User Data 5	1 Byte

NOTE: ensure test PC's system time to get right producing date.

Segment 2: VCM

VCM Start Flag and Start code LSB	1 Byte
VCM Start Code MSB	1 Byte
VCM End Flag and End code LSB	1 Byte
VCM End Code MSB	1 Byte

Confidential For Ebensz Only

Table - 10A OV4688 OTP Assignment

Address	Register name	Description
0x7000~ 0x710f	_	OmniVision reserved
Segment 1, Group 1		
0x7110	Flag1	Bit[7:6]: Flag of Group 1 00: Empty 01: Valid 1x: Invalid
0x7111	MID2	Bit[7:0]: Module Integrator ID of group 1 0x00: empty 0x01~0xff: Module integrator IDS Please contact with OmniVision FAE to get module integrator ID
0x7112	Lens ID1	Bit[7:0]: Lens ID of group 1
0x7113	YEAR1	Year of module production of group 1 e.g. 0x0c means year 2012
0x7114	MONTH1	Month of module production of group 1 e.g. 0x03 means march
0x7115	DAY1	Day of module production of group 1 e.g. 0x16 means Day 22th
0x7116	AWB_RG_MSB 1	MSB AWB R/G data of group 1 Bit[7:0] : AWB R/G ratio[9:2]
0x7117	AWB_BG_MSB 1	MSB AWB B/G data of group 1 Bit[7:0] : AWB B/G ratio[9:2]
0x7118	Light_RG_MSB 1	MSB Light source R/G data of group 1 Bit[7:0] : light source R/G ratio[9:2] Light source R/G ratio 0: 1 other : ([0x7118]<<2+ ([0x711A]&0x0C)>>2+512)/1024
0x7119	Light_BG_MSB 1	MSB Light source B/G data of group 1 Bit[7:0] : light source B/G ratio[9:2] Light source B/G ratio 0: 1 other : ([0x7119]<<2+[0x711A]&0x03+512)/1024
0x711A	AWB_Light_LSB 1	LSB for AWB data and light source of group 1 Bit[7:6] : AWB R/G ratio[1:0] Bit[5:4] : AWB B/G ratio[1:0] Bit[3:2] : Light source R/G ratio[1:0]

Confidential For EensZ Only

		Bit[2:1] : Light source B/G ratio[1:0]
0x711B	USER DATA 1-1	User Data 1 for customer of group 1
0x711C	USER DATA 1-2	User Data 2 for customer of group 1
0x711D	USER DATA 1-3	User Data 3 for customer of group 1
0x711E	USER DATA 1-4	User Data 4 for customer of group 1
0x711F	USER DATA 1-5	User Data 5 for customer of group 1
Segment 1, Group 2		
0x7120	MID2	Bit[7:0]: Module Integrator ID of group 2 0x00: empty 0x01~0xff: Module integrator IDS Please contact with OmniVision FAE to get module integrator ID
0x7121	Lens ID2	Bit[7:0]: Lens ID of group 2
0x7122	YEAR2	Year of module production of group 2 e.g. 0x0c means year 2012
0x7123	MONTH2	Month of module production of group 2 e.g. 0x03 means march
0x7124	DAY2	Day of module production of group 2 e.g. 0x16 means Day 22th
0x7125	DAY2	Day of module production of group 2 e.g. 0x16 means Day 22th
0x7126	AWB_RG_MSB 2	MSB AWB R/G data of group 2 Bit[7:0] : AWB R/G ratio[9:2]
0x7127	AWB_BG_MSB 2	MSB AWB B/G data of group 2 Bit[7:0] : AWB B/G ratio[9:2]
0x7128	Light_RG_MSB 2	MSB Light source R/G data of group 2 Bit[7:0] : light source R/G ratio[9:2] Light source R/G ratio 0: 1 other : ([0x7118]<<2+ ([0x711A]&0x0C)>>2+512)/1024
0x7129	Light_BG_MSB 2	MSB Light source B/G data of group 2 Bit[7:0] : light source B/G ratio[9:2] Light source B/G ratio 0: 1 other : ([0x7119]<<2+[0x711A]&0x03+512)/1024
0x712A	AWB_Light_LSB 2	LSB for AWB data and light source of group 2 Bit[7:6] : AWB R/G ratio[1:0]

Confidential For Ebenz Only

		Bit[5:4] : AWB B/G ratio[1:0] Bit[3:2] : Light source R/G ratio[1:0] Bit[2:1] : Light source B/G ratio[1:0]
0x712B	USER DATA 2-1	User Data 1 for customer of group 2
0x712C	USER DATA 2-2	User Data 2 for customer of group 2
0x712D	USER DATA 2-3	User Data 3 for customer of group 2
0x712E	USER DATA 2-4	User Data 4 for customer of group 2
0x712F	USER DATA 2-5	User Data 5 for customer of group 2
Segment 1, Group 3		
0x7130	MID3	Bit[7:0]: Module Integrator ID of group 3 0x00: empty 0x01~0xff: Module integrator IDS Please contact with OmniVision FAE to get module integrator ID
0x7131	Lens ID3	Bit[7:0]: Lens ID of group 3
0x7132	YEAR3	Year of module production of group 3 e.g. 0x0c means year 2012
0x7133	MONTH3	Month of module production of group 3 e.g. 0x03 means march
0x7134	DAY3	Day of module production of group 3 e.g. 0x16 means Day 22th
0x7135	DAY3	Day of module production of group 3 e.g. 0x16 means Day 22th
0x7136	AWB_RG_MSB 3	MSB AWB R/G data of group 3 Bit[7:0] : AWB R/G ratio[9:2]
0x7137	AWB_BG_MSB 3	MSB AWB B/G data of group 3 Bit[7:0] : AWB B/G ratio[9:2]
0x7138	Light_RG_MSB 3	MSB Light source R/G data of group 3 Bit[7:0] : light source R/G ratio[9:2] Light source R/G ratio 0: 1 other : ([0x7118]<<2+ ([0x711A]&0x0C)>>2+512)/1024
0x7139	Light_BG_MSB 3	MSB Light source B/G data of group 3 Bit[7:0] : light source B/G ratio[9:2] Light source B/G ratio 0: 1 other : ([0x7119]<<2+[0x711A]&0x03+512)/1024

Confidential For Ebensz Only

0x713A	AWB_Light_LSB 3	LSB for AWB data and light source of group 3 Bit[7:6] : AWB R/G ratio[1:0] Bit[5:4] : AWB B/G ratio[1:0] Bit[3:2] : Light source R/G ratio[1:0] Bit[2:1] : Light source B/G ratio[1:0]
0x713B	USER DATA 3-1	User Data 1 for customer of group 3
0x713C	USER DATA 3-2	User Data 2 for customer of group 3
0x713D	USER DATA 3-3	User Data 3 for customer of group 3
0x713E	USER DATA 3-4	User Data 4 for customer of group 3
0x713F	USER DATA 3-5	User Data 5 for customer of group 3

Confidential For Ebensz Only

Table – 10B OV4688 OTP assignment

Address	Register name	Description
Segment 2, Group 1		
0x7140	FLAGS1	Bit[7:6]: Flag of VCM start code of group 1 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0]: VCM start code LSB, VCMST[1:0] of group 1
0x7141	VCMST1	VCM start code MSB, VCMST[9:2] of group 1
0x7142	FLAGE1	Bit[7:6]: Flag of VCM end code of group 1 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0] VCM end code LSB, VCMED[1:0] of group 1
0x7143	VCMED1	VCM end code MSB, VCMED[9:2] of group 1
Segment 2, Group 2		
0x7144	FLAGS2	Bit[7:6]: Flag of VCM start code of group 2 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0]: VCM start code LSB, VCMST[1:0] of group 2
0x7145	VCMST2	VCM start code MSB, VCMST[9:2] of group 2
0x7146	FLAGE2	Bit[7:6]: Flag of VCM end code of group 2 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0] VCM end code LSB, VCMED[1:0] of group 2
0x7147	VCMED2	VCM end code MSB, VCMED[9:2] of group 2
Segment 2, Group 3		

Confidential For Ebensz Only

0x7148	FLAGS3	Bit[7:6]: Flag of VCM start code of group 3 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0]: VCM start code LSB, VCMST[1:0] of group 3
0x7149	VCMST3	VCM start code MSB, VCMST[9:2] of group 3
0x714A	FLAGE3	Bit[7:6]: Flag of VCM end code of group 3 00: empty 01: valid group 11: invalid group Bit[5:2]: Reserved Bit[1:0] VCM end code LSB, VCMED[1:0] of group 3
0x714B	VCMED3	VCM end code MSB, VCMED[9:2] of group 3

The 3 group data could be used for camera module rework. For the 1st time, calibration data and other data are written into group 1. If the camera module need to rework, then flag of group 1 is set to '1' to indicate that the data in group 1 is invalid. Then the new data could be written into group 2. If the camera is reworked for 2nd time, then flag of group 2 is set to '1' to indicate that data in group 2 is not valid and new data is written into group 3.

Confidential For Ebensz Only

5.2.1 Load Calibration Data

The calibration data of OV4688 camera module should be loaded after initialization (Global setting). Below is reference code for loading calibration data in camera driver. The code should be placed after camera module initialization.

Note:

OTPDPC function should be disabled by set register 0x5000[5] to '0' before access OTP. After OTP access, OTP DPC function should be turned on by set register 0x5000[5] to '1'.

```
struct otp_struct {
        int module_integrator_id;
        int lens_id;
        int production_year;
        int production_month;
        int production_day;
        int rg_ratio;
        int bg_ratio;
        int light_rg;
        int light_bg;
        int user_data[5];
```

```
	int VCM_start;
	int VCM_end;
};

// call this function before access OTP
int OV4688_OTP_access_start()
{
	int temp;

	temp = OV4688_read_i2c(0x5000);
	temp = temp & 0xdf;				// set bit 5 to '0'
	OV4688_write_i2c(0x5000, temp);

	return 0;
}

// call this function after access OTP
int OV4688_OTP_access_end()
{
	int temp;

	temp = OV4688_read_i2c(0x5000);
	temp = temp | 0x20;				// set bit 5 to '1'
	OV4688_write_i2c(0x5000, temp);

	return 0;
}

// index: index of otp group. (1, 2, 3)
// return:  0, group index is empty
//		1, group index has invalid data
//		2, group index has valid data
int check_otp_wb(int index)
{
	int flag, i;
	int address_start, address_end;
	if(index==1) {
		address_start = 0x7110;
		address_end = 0x7110;
	}
	else if(index==2) {
		address_start = 0x7120;
		address_end = 0x7120;
	}
	else {
		address_start = 0x7130;
		address_end = 0x7130;
	}

	// read otp into buffer
	OV4688_OTP_access_start();
	OV4688_write_i2c(0x3d84, 0xc0);	// program disable, manual mode

	//partial mode OTP write start address
	OV4688_write_i2c(0x3d88, (address_start>>8));
	OV4688_write_i2c(0x3d89, (address_start & 0xff));
	// partial mode OTP write end address
	OV4688_write_i2c(0x3d8A, (address_end>>8));
	OV4688_write_i2c(0x3d8B, (address_end & 0xff));

	OV4688_write_i2c(0x3d81, 0x01);	// read OTP
	Delay(5);

	//select group
```

Confidential For Ebensz Only

```
	flag = OV4688_read_i2c(address_start);

	// clear otp buffer
	for (i=address_start;i<=address_end;i++) {
		OV4688_write_i2c(i, 0x00);
	}
	OV4688_OTP_access_end();

	if (flag == 0x00) {
		return 0;
	}
	else if (flag & 0x02) {
		return 1;
	}
	else {
		return 2;
	}
}

// index: index of otp group. (1, 2, 3)
// code: 0 for start, 1 for end
// return:  0, group index is empty
//		1, group index has invalid data
//		2, group index has valid data
int check_otp_VCM(int index, int code)
{
	int flag, i;
	int address_start, address_end;

	if(index=1) {
		address_start = 0x7140 + code*2;
		address_end = 0x7140 + code*2;
	}
	else if(index=2) {
		address_start = 0x7144 + code*2;
		address_end = 0x7144 + code*2;
	}
	else {
		address_start = 0x7148 + code*2;
		address_end = 0x7148 + code*2;
	}

	// read otp into buffer
	OV4688_OTP_access_start();
	OV4688_write_i2c(0x3d84, 0xc0);	// program disable, manual mode

	//partial mode OTP write start address
	OV4688_write_i2c(0x3d88, (address_start>>8));
	OV4688_write_i2c(0x3d89, (address_start & 0xff));
	// partial mode OTP write end address
	OV4688_write_i2c(0x3d8A, (address_end>>8));
	OV4688_write_i2c(0x3d8B, (address_end & 0xff));

	OV4688_write_i2c(0x3d81, 0x01);
	Delay(5);

	//select group
	flag = OV4688_read_i2c(address_start);
	flag = flag & 0xc0;

	// clear otp buffer
	for (i=address_start;i<=address_end;i++) {
		OV4688_write_i2c(i, 0x00);
	}
```

Confidential For Ebensz Only

```
        OV4688_OTP_access_end();

        if (flag == 0x00) {
                return 0;
        }
        else if (flag & 0x80) {
                return 1;
        }
        else {
                return 2;
        }
}


// index: index of otp group. (1, 2, 3)
// otp_ptr: pointer of otp_struct
// return:  0,
int read_otp_wb(int index, struct otp_struct *otp_ptr)
{
        int i, temp;
        int address_start, address_end;

        // read otp into buffer
        OV4688_write_i2c(0x3d84, 0xc0);      // program disable, manual mode

        //select group
        if(index==1)
        {
           address_start = 0x7111;
                address_end = 0x711f;
        }
        else if(index==2)
        {
           address_start = 0x7121;
                address_end = 0x712f;
        }
        else
        {
           address_start = 0x7131;
                address_end = 0x713f;
        }

        //partial mode OTP write start address
        OV4688_OTP_access_start();
        OV4688_write_i2c(0x3d88, (address_start>>8));
        OV4688_write_i2c(0x3d89, (address_start & 0xff));
        // partial mode OTP write end address
        OV4688_write_i2c(0x3d8A, (address_end>>8));
        OV4688_write_i2c(0x3d8B, (address_end & 0xff));

        OV4688_write_i2c(0x3d81, 0x01);      // load otp into buffer
        Delay(5);

        (*otp_ptr).module_integrator_id = OV4688_read_i2c(address_start);
        (*otp_ptr).lens_id = OV4688_read_i2c(address_start + 1);
        (*otp_ptr).production_year = OV4688_read_i2c(address_start + 2);
        (*otp_ptr).production_month = OV4688_read_i2c( address_start + 3);
        (*otp_ptr).production_day = OV4688_read_i2c(address_start + 4);
        temp = OV4688_read_i2c(address_start + 9);
        (*otp_ptr).rg_ratio = (OV4688_read_i2c(address_start + 5)<<2) + ((temp>>6) & 0x03);
        (*otp_ptr).bg_ratio = (OV4688_read_i2c(address_start + 6)<<2) + ((temp>>4) & 0x03);
        (*otp_ptr).light_rg = (OV4688_read_i2c(address_start + 7) <<2) + ((temp>>2) & 0x03);
        (*otp_ptr).light_bg = (OV4688_read_i2c(address_start + 8)<<2) +  (temp & 0x03);
        (*otp_ptr).user_data[0] = OV4688_read_i2c(address_start + 10);
```

Confidential For Ebensz Only

```
	(*otp_ptr).user_data[1] = OV4688_read_i2c(address_start + 11);
	(*otp_ptr).user_data[2] = OV4688_read_i2c(address_start + 12);
	(*otp_ptr).user_data[3] = OV4688_read_i2c(address_start + 13);
	(*otp_ptr).user_data[4] = OV4688_read_i2c(address_start + 14);

	// clear otp buffer
	for (i=address_start;i<=address_end;i++) {
		OV4688_write_i2c(i, 0x00);
	}
	OV4688_OTP_access_end();

	return 0;
}

// index: index of otp group. (1, 2, 3)
// code: 0 start code, 1 stop code
// return:  0
int read_otp_VCM(int index, int code, struct otp_struct * otp_ptr)
{
	int i, temp, vcm;
	int address;
	int address_start, address_end;

	//check group
	if(index==1)
	{
	  address_start = 0x7140;
		address_end = 0x7143;
	}
	else if(index==2)
	{
	  address_start = 0x7144;
		address_end = 0x7147;
	}
	else
	{
	  address_start = 0x7148;
		address_end = 0x714b;
	}

	// read otp into buffer
	OV4688_OTP_access_start();
	OV4688_write_i2c(0x3d84, 0xc0);	// program disable, manual mode

	//partial mode OTP write start address
	OV4688_write_i2c(0x3d88, (address_start>>8));
	OV4688_write_i2c(0x3d89, (address_start & 0xff));
	// partial mode OTP write end address
	OV4688_write_i2c(0x3d8A, (address_end>>8));
	OV4688_write_i2c(0x3d8B, (address_end & 0xff));

	OV4688_write_i2c(0x3d81, 0x01);	// load otp into buffer
	Delay(5);

	//flag and lsb of VCM start code
	address = address_start + code*2;
	temp = OV4688_read_i2c(address);
	vcm = (OV4688_read_i2c(address_start + 1) << 2) | (temp & 0x03);

	if(code==1) {
		(* otp_ptr).VCM_end = vcm;
	}
	else {
		(* otp_ptr).VCM_start = vcm;
```

Confidential For Ebensz Only

```
	}

	// clear otp buffer
	for (i=address_start;i<=address_end;i++) {
		OV4688_write_i2c(i, 0x00);
	}
	OV4688_OTP_access_end();

	return 0;
}

// R_gain, sensor red gain of AWB, 0x400 =1
// G_gain, sensor green gain of AWB, 0x400 =1
// B_gain, sensor blue gain of AWB, 0x400 =1
// return 0;
int update_awb_gain(int R_gain, int G_gain, int B_gain)
{
	if (R_gain>0x400) {
		//long WB R gain
		OV4688_write_i2c(0x500c, R_gain>>8);
		OV4688_write_i2c(0x500d, R_gain & 0x00ff);
		//middle WB R gain
		OV4688_write_i2c(0x5012, R_gain>>8);
		OV4688_write_i2c(0x5013, R_gain & 0x00ff);
		//short WB R gain
		OV4688_write_i2c(0x5018, R_gain>>8);
		OV4688_write_i2c(0x5019, R_gain & 0x00ff);
	}

	if (G_gain>0x400) {
		//long WB G gain
		OV4688_write_i2c(0x500e, G_gain>>8);
		OV4688_write_i2c(0x500f, G_gain & 0x00ff);
		//middle WB G gain
		OV4688_write_i2c(0x5014, G_gain>>8);
		OV4688_write_i2c(0x5015, G_gain & 0x00ff);
		//short WB G gain
		OV4688_write_i2c(0x501A, G_gain>>8);
		OV4688_write_i2c(0x501B, G_gain & 0x00ff);
	}

	if (B_gain>0x400) {
		//long WB B gain
		OV4688_write_i2c(0x5010, B_gain>>8);
		OV4688_write_i2c(0x5011, B_gain & 0x00ff);
		//middle WB B gain
		OV4688_write_i2c(0x5016, B_gain>>8);
		OV4688_write_i2c(0x5017, B_gain & 0x00ff);
		//short WB B gain
		OV4688_write_i2c(0x501C, B_gain>>8);
		OV4688_write_i2c(0x501D, B_gain & 0x00ff);
	}
	return 0;
}


// call this function after OV4688 initialization
// return value: 0 update success
//		1, no OTP
int update_otp_wb()
{
	struct otp_struct current_otp;
	int i;
	int otp_index;// bank 1,2,3
```

Confidential For Ebensz Only

```
int temp;
int R_gain, G_gain, B_gain, G_gain_R, G_gain_B;
int rg,bg;


// R/G and B/G of current camera module is read out from sensor OTP
// check first OTP with valid data
for(i=1;i<=3;i++) {
        temp = check_otp_wb(i);
        if (temp == 2) {
                otp_index = i;
                break;
        }
}

if (i>3) {
        // no valid wb OTP data
        return 1;
}


// set right bank

read_otp_wb(otp_index, &current_otp);

if(current_otp.light_rg==0) {
        // no light source information in OTP, light factor = 1
        rg = current_otp.rg_ratio;
}
else {
        rg = current_otp.rg_ratio * (current_otp.light_rg + 512) / 1024;
}

if(current_otp.light_bg==0) {
        // not light source information in OTP, light factor = 1
        bg = current_otp.bg_ratio;
}
else {
        bg = current_otp.bg_ratio * (current_otp.light_bg + 512) / 1024;
}

//calculate G gain
//0x400 = 1x gain
if(bg < BG_Ratio_Typical) {
        if (rg< RG_Ratio_Typical) {
                // current_otp.bg_ratio < BG_Ratio_typical &&
                // current_otp.rg_ratio < RG_Ratio_typical
                G_gain = 0x400;
                B_gain = 0x400 * BG_Ratio_Typical / bg;
        R_gain = 0x400 * RG_Ratio_Typical / rg;
        }
        else {
                // current_otp.bg_ratio < BG_Ratio_typical &&
                // current_otp.rg_ratio >= RG_Ratio_typical
        R_gain = 0x400;
                G_gain = 0x400 * rg / RG_Ratio_Typical;
        B_gain = G_gain * BG_Ratio_Typical /bg;
        }
}
else {
        if (rg < RG_Ratio_Typical) {
                // current_otp.bg_ratio >= BG_Ratio_typical &&
                // current_otp.rg_ratio < RG_Ratio_typical
```

Confidential For Ebensz Only

```
                B_gain = 0x400;
                G_gain = 0x400 * bg / BG_Ratio_Typical;
                R_gain = G_gain * RG_Ratio_Typical / rg;
                }
                else {
                        // current_otp.bg_ratio >= BG_Ratio_typical &&
                        // current_otp.rg_ratio >= RG_Ratio_typical
                G_gain_B = 0x400 * bg / BG_Ratio_Typical;
                        G_gain_R = 0x400 * rg / RG_Ratio_Typical;

                if(G_gain_B > G_gain_R ) {
                                                B_gain = 0x400;
                                                G_gain = G_gain_B;
                                        R_gain = G_gain * RG_Ratio_Typical /rg;
                                }
                else {
                                        R_gain = 0x400;
                                        G_gain = G_gain_R;
                                        B_gain = G_gain * BG_Ratio_Typical / bg;
                                }
        }
        }

        update_awb_gain(R_gain, G_gain, B_gain);

        return 0;

}
```

After calibration applied, the R/G and B/G of camera module is same or very close to R/G and B/G of typical camera module. The lens correction output of camera module is same or very close to lens correction output of typical camera module. Also the BLC value will be better than common BLC value.

5.3 Driver Functions

5.3.1 Exposure

The registers 0x3500 ,0x3501 and 0x3502 control exposure of OV4688. The exposure value is in number of Tline, where Tline is the time of sensor one line.

Exposure = [reg 0x3500]<<12 + [reg 0x3501]<<4 + [reg 0x3502]>>4;

The maximum exposure value is limited by VTS defined by register 0x380e and 0x380f.
Maximum Exposure <= VTS -4

```
int OV4688_get_shutter()
{
        // read shutter, in number of line period
        int shutter;

        shutter = (OV4688_read_i2c(0x03500) & 0x0f);
        shutter = (shutter<<8) + OV4688_read_i2c(0x3501);
        shutter = (shutter<<4) + (OV4688_read_i2c(0x3502)>>4);

        return shutter;
}

int OV4688_set_shutter(int shutter)
```

Confidential For Ebensz Only

```
{
        // write shutter, in number of line period
        int temp;

        shutter = shutter & 0xffff;

        temp = shutter & 0x0f;
        temp = temp<<4;
        OV4688_write_i2c(0x3502, temp);

        temp = shutter & 0xfff;
        temp = temp>>4;
        OV4688_write_i2c(0x3501, temp);

        temp = shutter>>12;
        OV4688_write_i2c(0x3500, temp);

        return 0;
}
```

If the exposure written to OV4688 exceed limit of VTS, then VTS need to be extended.

```
int OV4688_set_VTS(int VTS)
{
        // write VTS to registers
        int temp;

        temp = VTS & 0xff;
        OV4688_write_i2c(0x380f, temp);

        temp = VTS>>8;
        OV4688_write_i2c(0x380e, temp);

        return 0;
}

 // write capture shutter
if (capture_shutter > (capture_VTS - 4)) {
        capture_VTS = capture_shutter + 4;
        OV4688_set_VTS(capture_VTS);
}

OV4688_set_shutter(capture_shutter);
```

Confidential For Ebensz Only

OV4688 doesn't support AGC and AEC. For real application, customers need use AGC/AEC of back-end DSP.

5.3.2 Gain

Gain is controlled by register 0x3508 and 0x3509.

The range of gain is from 1.0 to 16.

```
int OV4688_get_gain128()
{
        // read gain, 128 = 1x
        int gain_regh, gain_regl, gain128;

        gain_regh = OV4688_read_i2c(0x3508) & 0x07;
        gain_regl = OV4688_read_i2c(0x3509);
```

```
        if(gain_regh == 0) {
                gain128 = gain_regl;
        }
        else if(gain_regh == 1) {
                gain128 = (gain_regl + 8)*2;
        }
        else if(gain_regh == 3) {
                gain128 = (gain_regl + 12)*4;
        }
        else if(gain_regh == 7) {
                gain128 = (gain_regl + 8)*8;
        }

        return gain128;
}

int OV4688_set_gain128(int gain128)
{
        // write gain, 128 = 1x
        int gain_regh, gain_regl;

        gain128 = gain128 & 0x7ff;

        if(gain128 < 256) {
                gain_regh = 0;
                gain_regl = gain128;
        }
        else if(gain128 < 512) {
                gain_regh = 1;
                gain_regl = gain128/2 - 8;
        }
        else if(gain128 < 1024) {
                gain_regh = 3;
                gain_regl = gain128/4 - 12;
        }
        else {
                gain_regh = 7;
                gain_regl = gain128/8 - 8;
        }

        OV4688_write_i2c(0x3508, gain_regh);
        OV4688_write_i2c(0x3509, gain_regl);

        return 0;
}
```

Confidential For Ebensz Only

5.3.3 Capture Sequence

```
        int preview_shutter, preview_gain128, preview_SCLK, preview_HTS, preview_binning;
        int capture_shutter, capture_gain128, capture_SCLK, capture_HTS, capture_VTS;
        long capture_gain128_shutter;
        int XVCLK = 2400;

        // read preview shutter
        preview_shutter = OV4688_get_shutter();

        // read preview gain
        preview_gain128 = OV4688_get_gain128();

        // read preview PCLK
        preview_SCLK = OV4688_get_SCLK(XVCLK);
```

```
	// read preview HTS
	preview_HTS = OV4688_get_HTS();

	// get preview binning factor
	preview_binning = OV4688_get_binning_factor();

	// Write full resolution 16:9 table
	for (int i=0; i<sizeof(regFull169)/sizeof(int); i+=2)
	{
		OV4688_write_i2c(regFull169[i], regFull169[i+1]);
	}

	// read capture PCLK
	capture_SCLK = OV4688_get_SCLK(XVCLK);

	// read capture HTS
	capture_HTS = OV4688_get_HTS();

	// read capture VTS
	capture_VTS = OV4688_get_VTS();

	// calculate capture shutter
	capture_shutter = preview_shutter * capture_SCLK/preview_SCLK * preview_HTS/capture_HTS * preview_binning;

	if(capture_shutter<16) {
		capture_gain128_shutter = preview_shutter * preview_gain128 * capture_SCLK/preview_SCLK *
preview_HTS / capture_HTS;
		capture_shutter = int(capture_gain128_shutter / 128);
		if(capture_shutter <1)
			capture_shutter = 1;

		capture_gain128 = int(capture_gain128_shutter / capture_shutter);
		if(capture_gain128<128)
			capture_gain128 = 128;
	}
	else {
		capture_gain128 = preview_gain128;
	}

	// gain to shutter
	while ((capture_gain128>256) && (capture_shutter < int((capture_VTS-4)/2))) {
		// capture gain > 2x, capture shutter < 1/2 max shutter
		capture_gain128 = capture_gain128 / 2;
		capture_shutter = capture_shutter * 2;
	}

	// write capture gain
	OV4688_set_gain128(capture_gain128);

	// write capture shutter
	if (capture_shutter > (capture_VTS - 6)) {
		capture_VTS = capture_shutter + 6;
		OV4688_set_VTS(capture_VTS);
	}
	OV4688_set_shutter(capture_shutter);

	// skip 2 vysnc

	// start capture at 3rd vsync
```

Confidential For Ebensz Only

5.3.5 SCCB Group Write

TBA.

5.3.6 Capture with Flash LED

TBA

5.4 Verification of Camera Driver Functions

To verify function of camera driver, please make sure that

1. OV4688_get_PCLK() return right values of PCLK

2. Try to capture picture in dark, normal and bright condition, check the brightness of pictures captured is correct and there is no light band in captured picture.

Confidential For Ebensz Only

6. Tuning Image Quality

6.1 Using Calibration Data

Before image quality tuning, please make sure that calibration data of OV4688 camera module is successfully loaded in camera driver and updated after initial settings of OV4688.

6.2 Find Typical Camera Module and Corner Camera Modules

Prepare 50 pcs camera modules, capture raw picture under 5100K light of DNP viewer. Analyze R/G, B/G and lens shading with camera analyzer. Find typical camera modules for tuning and corner camera modules for tuning result check.

6.3 Tune with ISP Tools

Capture pictures with typical camera module. Tune settings with ISP tools. Implement settings in camera driver and ISP driver.

6.4 Verify Settings Tuned

Verify settings tuned with both typical camera module and corner camera modules.

Confidential For Ebensz Only

Revision History

Rev 1.00

Initial Release. Modified from “OV8830 Camera Module Application Notes R1.22”. Settings follow “OV4688_AM19.ovd”

Rev 1.01

Update register settings following “OV4688_AM19F.ovd”.

Rev 1.02

Update OTP definition and sample code to be consistent with “OV4688 Calibration and OTP Programming Guide R2.0”.

Rev 1.03

Update table - 3 Power supply requiremnt.

3.1.2 Clock “less than 5ns” changed to “less than 27% of whole clock period”.

3.1.6 PCB Layout Guide of Hose, remove “DOVDD” from 3. Add 4 and 5.

Update Table – 6 Comparison of hardware operation schemes

Update 4.1.2 Enter Camera

Update 4.1.3 Switch to other camera

Update 4.1.4 Switch from other camera

Update 4.1.5 Exit camera

Update 4.2.1 Sysem power on

Update 4.2.2 Enter Camera

Update 4.2.3 Switch to other camera

Update 4.2.4 Switch from other camera

5.1.1 Initialization, change delay from 2ms to 10ms

5.7 Mirror and Flip, add note

5.8 BLC strategy, change offset our of range trigger to off, add note.

5.2.1 Load calibration data, add note, add function OV4688_OTP_access_start() and OV4688_OTP_access_end(); Update function check_otp_wb(); check_otp_VCM(); read_otp_WB(); read_otp_VCM();

Rev 1.04

Confidential For Ebensz Only

Update 3.1.1 Power Supplies.

Confidential For Ebensz Only