Crystal Image through
Imaging Innovation
PIXELPLUS

Preliminary
Datasheet

1/4 inch Bayer Chip
CMOS Image Sensor with 1320x988 Pixel Array

PK3130K

Rev 0.8

Last update : 13. Jul. 2020

6th Floor, Gyeonggi R&DB Center, 906-5 Iui-dong, Yeongtong-gu,
Suwon-si, Gyeonggi-do, 443-766, Korea
Tel : 82-31-888-5300, FAX : 82-31-888-5398

Table of Contents

List of Figures

List of Tables

Features

- 1320 x 988 effective pixel array with RGB bayer color filters and micro-lens
- Output format: RGB Bayer (progressive scan)
- Interface
  - 12-bit digital parallel interface support
  - 2/4-lane serial output MIPI support @ max 1.04[Gbps/lane]
- High dynamic range
- HDR combine
- Auto black level compensation
- Programmable frame size, frame rate, window size, image orientation, and exposure
- Horizontal/Vertical mirroring and image cropping
- External synchronization support (Genlock)
- Chip address selection PAD
- Software reset
- On-chip phase locked loop (PLL)
- I2C Interface support

General Description

The PK3130K is a 1/4-inch CMOS image sensor with high dynamic range. It is a Bayer sensor with effective pixel array of 1320 (width) x 988 (height). The PK3130K can generate a 12-bit RGB raw Bayer data at maximum frame rate of 30 FPS through MIPI serial interface or 12-bit parallel interface. On-chip sensor functions can be controlled through I2C interface.

Table 1 Key Performance Parameter

Parameter	Typical value
Pixel size	3 [um] x 3 [um]
Effective pixel array	1320 (H) x 988 (V)
Effective image area	3.960 [mm] x 2.964 [mm]
Optical format	1/4 [inch]
Input clock frequency	27 [MHz]
Output interface	12-bit parallel
	MIPI serial interface with 2/4 lane
Max. frame rate	30 [FPS]
Dark signal	6.6 [mV/sec] @60 [°C]

Parameter	Typical value
Sensitivity	32.5k[e-/Lux.sec]
Power supply	HVDD : 3.3 or 1.8 [V]
	AVDD : 3.3 [V]
	DVDD : 1.2 [V]
Power consumption	149.4 [mW] @dynamic (Pararrel)
	134.1 [mW] @dynamic (MIPI)
	183.6 [uW] @standby
Operating Temp. (fully functional Temp.)	-40~85 [°C] (Ambient)
Dynamic range	100 [dB]
SNR	41.4 [dB]
Package Type	CSP

Chip Architecture

The PK3130K has a 1320 x 988 effective pixel array and includes column/row driver circuits for reading out pixel data progressively. CDS circuit reduces noises generated from various sources, which mainly are resulted from process variations. The fixed errror signal level caused by pixel process variation can be reduced by sampling the defference between the output and the reset level of the pixel. Each of R, G, and B pixel output can be multiplied by different gain factors to balance the color of images under various light conditions. The analog signals are converted into digital data of one line at a time and each line data is streamed out column by column. The Bayer RGB data passes through a sequence of image signal processing to produce bayer output data. The PK3130K supports output interfaces such as a 12-bit parallel and the MIPI. Control of internal functions or output signal timings can be enabled by modifying registers directly through a 2-wire serial interface called I2C.

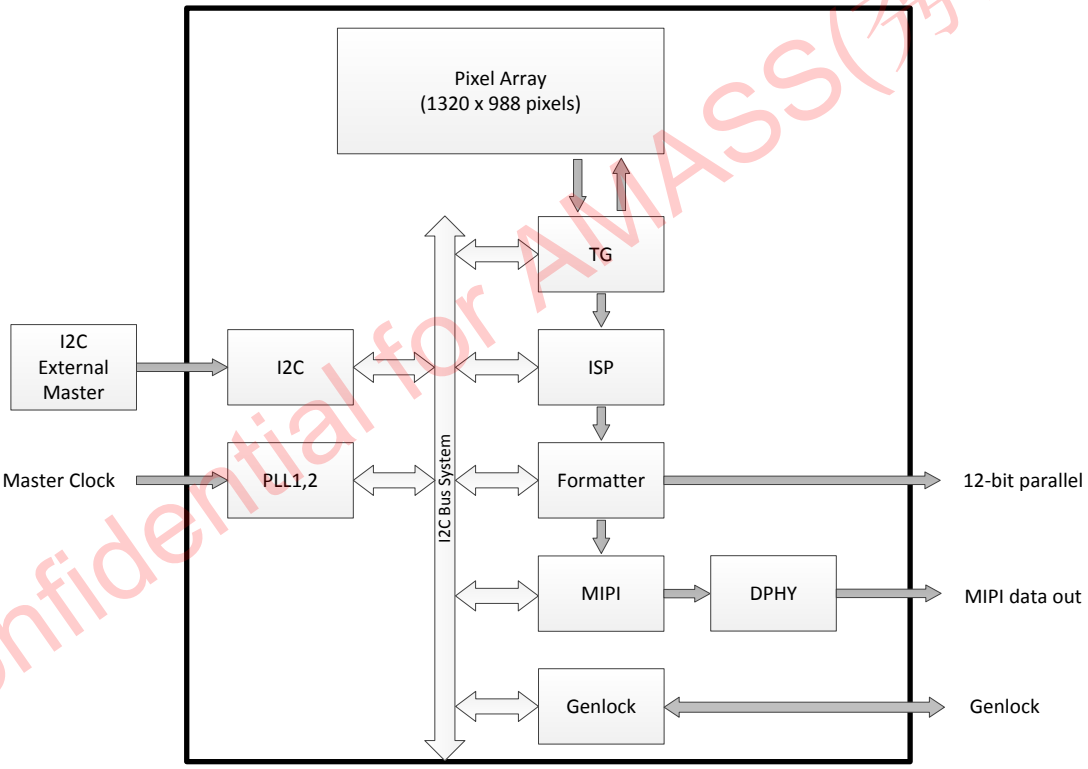


Figure 1 Chip architecture

Frame Structure

The size of a frame is determined by framewidth and frameheight registers. One frame consists of (framewidth + 1) columns and (frameheight + 1) rows, where the size of one frame is allowed to be larger than the total pixel array size. Window determines the output image size, and its default size is 1296 x 976 pixels. It is possible to define a specific region of the frame by a determined window. Pixel scanning is performed row by row on entire frame. Frame row counter and frame column counter, which are limited by framewidth and frameheight values respectively, are used to indicate the current coordinate of pixel being scanning. The column counter value increase by every pixel clock (pclk). every time the column counter reaches maximum value, the row counter value increase. Figure 2 shows the default frame structure and the window position of the PK3130K with origin point (0,0) in the top right corner. Figure 3 shows a 720p frame structure.

As shown in Figure 3, the cropping function can reduce the frame height according to the register value set by users and it enables the frame rate to increase depending on the reduction of frame size. The amount of cropping can be controlled by rjump_top, rjump_bot,and rjump_blank registers. These registers control the crop size of top frame, bottom frame, and blank frame respectively.

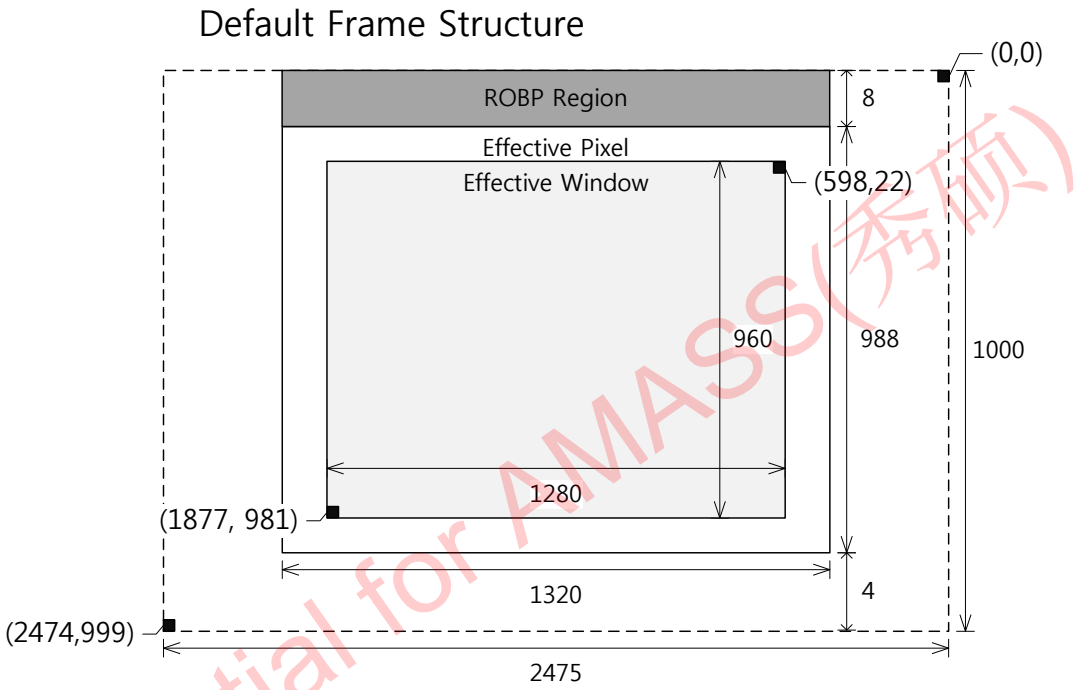


Figure 2 Default frame structure(top view)

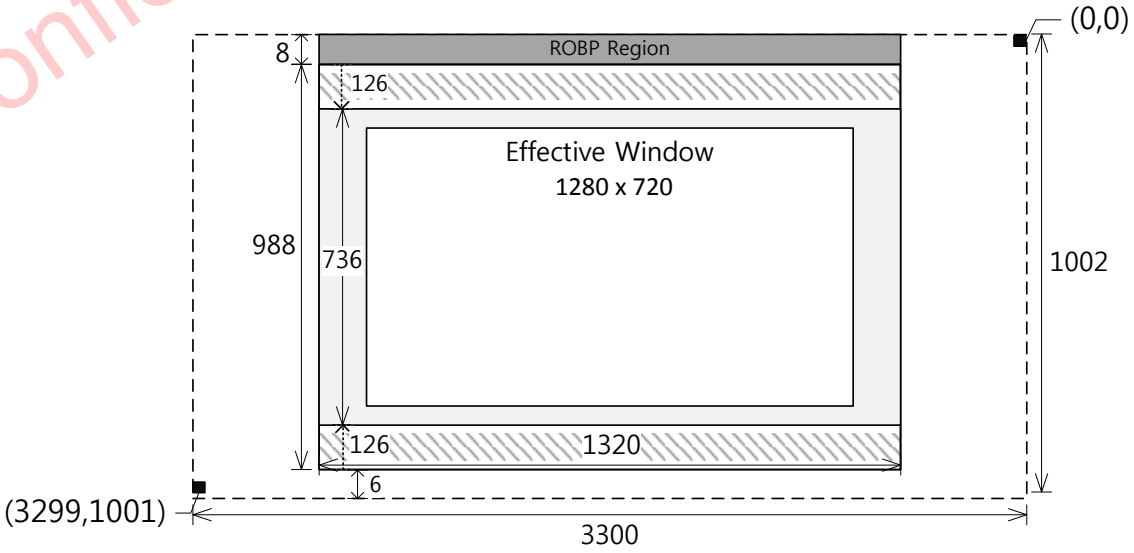


Figure 3 720p frame structure

Table 2 Register Table - Frame structure

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
framewidth_h	A	06	[4:0]	0x0B	RW	aev	Framewidth High Byte (must be larger than window width)
framewidth_l	A	07	[7:0]	0x99	RW	aev	Framewidth Low Byte (must be larger than window width)
fheight_a_h	A	08	[4:0]	0x03	RW	aev	Frameheight High Byte (must be larger than window height)
fheight_a_l	A	09	[7:0]	0xE7	RW	aev	Frameheight Low Byte (must be larger than window height)

Pixel Data Format

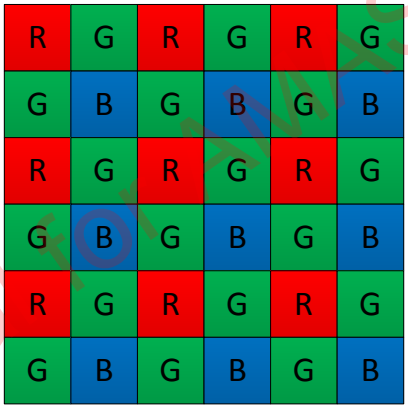


Figure 4 Bayer color filter pattern

The pixel array is covered by Bayer color filters as shown in Figure 4. Since each pixel can have only one type of filter on it, only one color component can be produced by a pixel. The PK3130K provides RGB Bayer pattern data through a 10-bit channel which it passes one pixel data to the output bus at every pclk.

The PK3130K provides horizontal, vertical mirror which respectively reverse the sensor data readout order horizontaly and vertically. Figure 5 shows a normal image and a mirrored image.

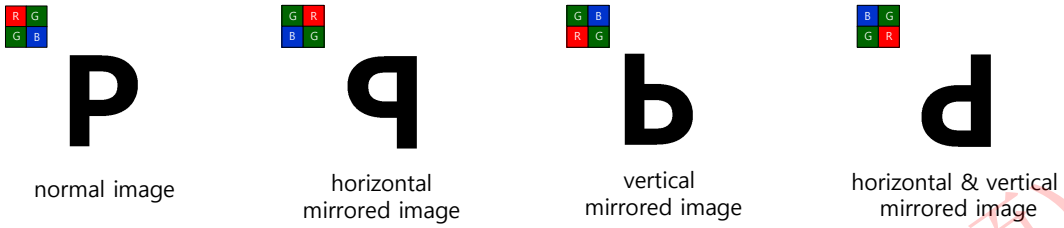


Figure 5 Mirror

Table 3 shows registers relevant to mirror.

Table 3 Register Table - Mirror

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mirror	A	05	[1:0]	0x00	RW	aev	Image Inversion mirror[1] : vertical inversion mirror[0] : horizontal inversion

Parallel Formatter

The parallel data is controlled by format header, also called timing reference sequence (TRS). The TRS indicates Start or End of video and is included with pixel data during serial transfer. Figure 6 shows TRS and vertical timing

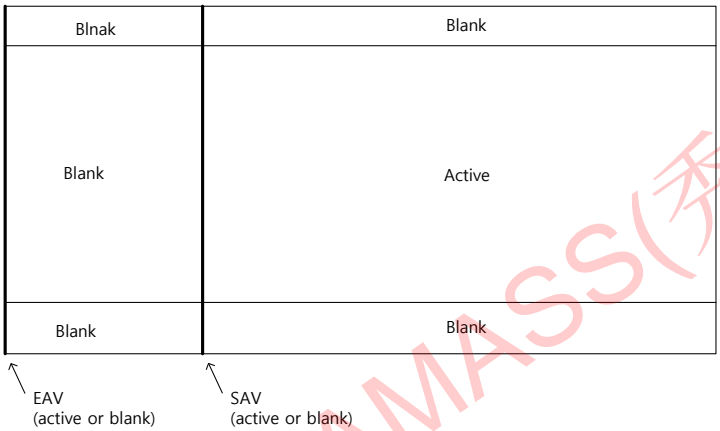


Figure 6 Parallel format

Active EAV, active SAV, blank EAV, blank SAV and blank data shown in Figure 6 are generated as follows:
SAV = {sync_CCIR_FF, sync_CCIR_00, sync_CCIR_00, sync_blankSAV}
EAV = {sync_CCIR_FF, sync_CCIR_00, sync_CCIR_00, sync_blankEAV}
aSAV = {sync_CCIR_FF, sync_CCIR_00, sync_CCIR_00, sync_activeSAV}
aEAV = {sync_CCIR_FF, sync_CCIR_00, sync_CCIR_00, sync_activeEAV}
BLANK= {sync_CCIR_80, sync_CCIR_10} ••• {sync_CCIR_80, sync_CCIR_10}

Table 4 shows registers relevant to parallel format.

Table 4 Register Table - Parallel format

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
sync_blankEAV_h	A	58	[3:0]	0x0B	RW		Blanking EAV control High Byte
sync_blankEAV_l	A	59	[7:0]	0x60	RW		Blanking EAV control Low Byte
sync_blankSAV_h	A	5A	[3:0]	0x0A	RW		Blanking SAV control High Byte
sync_blankSAV_l	A	5B	[7:0]	0xB0	RW		Blanking SAV control Low Byte
sync_activeEAV_h	A	5C	[3:0]	0x09	RW		Active EAV control High Byte
sync_activeEAV_l	A	5D	[7:0]	0xD0	RW		Active EAV control Low Byte
sync_activeSAV_h	A	5E	[3:0]	0x08	RW		Active SAV control High Byte
sync_activeSAV_l	A	5F	[7:0]	0x00	RW		Active SAV control Low Byte
sync_CCIR_FF_h	A	60	[3:0]	0x0F	RW		Format header control 0 (FF) High Byte
sync_CCIR_FF_l	A	61	[7:0]	0xFF	RW		Format header control 0 (FF) Low Byte
sync_CCIR_00_h	A	62	[3:0]	0x00	RW		Format header control 1 (00) High Byte
sync_CCIR_00_l	A	63	[7:0]	0x00	RW		Format header control 1 (00) Low Byte

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
sync_CCIR_80_h	A	64	[3:0]	0x08	RW		Format header control 2 (80) High Byte
sync_CCIR_80_l	A	65	[7:0]	0x00	RW		Format header control 2 (80) Low Byte
sync_CCIR_10_h	A	66	[3:0]	0x01	RW		Format header control 3 (10) High Byte
sync_CCIR_10_l	A	67	[7:0]	0x00	RW		Format header control 3 (10) Low Byte

When data_clamp is enabled, active data is clamped by data_min and data_max as shown in Table 5. data_min determine minimum value of active data, and data_max determine maximum value of active data.

Table 5 Register Table - Active data(data_clamp = enable)

output bit	data_min	data_max
MSB 8bit	010h	FE0h
MSB 9bit	008h	FF0h
MSB 10bit	004h	FF8h
MSB 11bit	002h	FFCh
MSB 12bit	001h	FFEh

Table 6 shows registers relevant to data clamp.

Table 6 Register Table - Data clamp

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
data_clamp	A	87	[1]	1'b0	RW	aev	Effective data clamping enable 1'b0 : disable 1'b1 : enable
data_min_h	A	8E	[3:0]	0x00	RW		Active data control High Byte data_clamp = 1'b1 acive data < data_min output data = data_min
data_min_l	A	8F	[7:0]	0x01	RW		Active data control Low Byte data_clamp = 1'b1 acive data < data_min output data = data_min
data_max_h	A	90	[3:0]	0x00	RW		Active data control High Byte data_clamp = 1'b1 acive data > data_max output data = data_max
data_max_l	A	91	[7:0]	0xFE	RW		Active data control Low Byte data_clamp = 1'b1 acive data > data_max output data = data_max

By manipulating vsyncstartrow0, vsyncstoprow0, and vsynccolumn0 register value, start and stop positions of vsync are controlled. sync_drop register allows user to drop vsync or hsync. Figure 7 shows 4 different cases of sync_drop. In addition, sync_hsyncAllLines enables hsync during vsync blank region. Figure 8 shows operation of sync_hsyncAlllines.

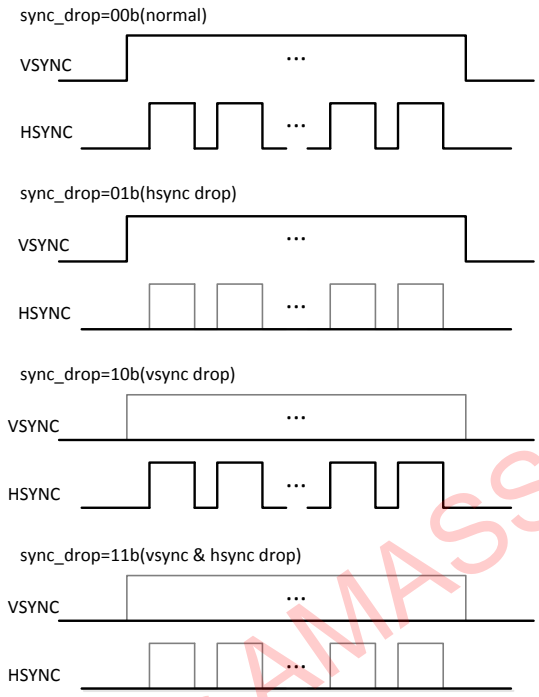


Figure 7 Sync drop

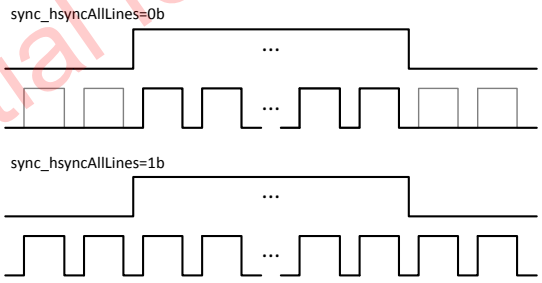


Figure 8 Hsync all lines

sync_vsyncPolarity, sync_hsyncPolarity registers invert vsync, hsync signal respectively. The inversion functions are shown in Figure 9, Figure 10.

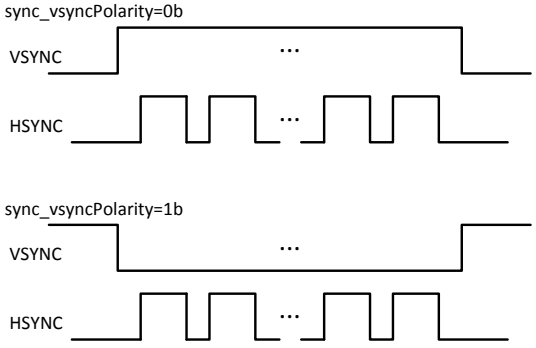


Figure 9 Vsync polarity

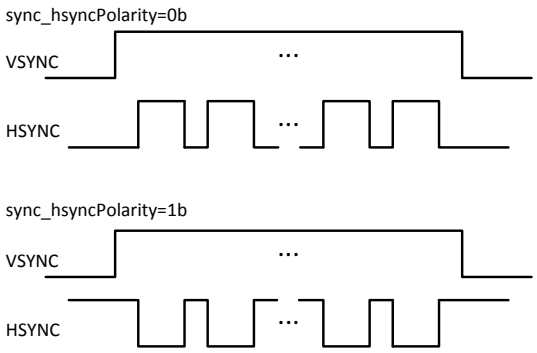


Figure 10 Hsync polarity

Table 7 Register Table - Sync control

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
vsyncstartrow0_h	A	69	[4:0]	0x00	RW	aev	Vertical sync start control for parallel interface High Byte Frame start control for MIPI interface High Byte
vsyncstartrow0_l	A	6A	[7:0]	0x0D	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsyncstoprow0_h	A	6B	[4:0]	0x03	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsyncstoprow0_l	A	6C	[7:0]	0xDD	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsynccolumn0_h	A	6D	[4:0]	0x00	RW	aev	Internal vsync 0 start point High Byte @ column counter
vsynccolumn0_l	A	6E	[7:0]	0x02	RW	aev	Internal vsync 0 start point Low Byte @ column counter
sync_drop[1:0]	A	86	[6:5]	2'b00	RW	aev	Vsync, hsync drop control 2'b00 : No drop 2'b01 : vsync drop 2'b10 : hsync drop 2'b11 : hsync and vsync drop
sync_vsyncPolarity	A	87	[6]	1'b0	RW	aev	Vsync polarity change 1'b0 : disable 1'b1 : enable
sync_hsyncPolarity	A	87	[4]	1'b0	RW	aev	Hsync polarity change 1'b0 : disable 1'b1 : enable
sync_hsyncAllLines	A	87	[5]	1'b0	RW	aev	Hsync output all lines enable(black and active) 1'b0 : No hsync during vertical blank 1'b1 : hsync during vertical blank

Digital Parallel Interface

The digital parallel interface uses VSYNC, HSYNC, PCLK, D[11:0] PIN. Table 8 shows digital parallel interface control registers.

Table 8 Register Table - digital parallel interface

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
vsync_pad_en	A	29	[7]	1'b0	RW		Vsync pad enable 1'b0 : disable 1'b1 : enable
hsync_pad_en	A	29	[4]	1'b0	RW		Hsync pad enable 1'b0 : disable 1'b1 : enable
hsync_drv	A	29	[6:5]	2'b00	RW		Hsync Pad drivability control
dly_hsync_ctrl[2:0]	A	2C	[6:4]	3'b000	RW		HSYNC timing delay delay = dly_hsync_ctrl*0.8 ns
pclk_pad_en	A	29	[3]	1'b0	RW		PCLK pad enable 1'b0 : disable 1'b1 : enable
pclk_drv[1:0]	A	28	[5:4]	2'b00	RW		PCLK pad drivability control
pclk_polarity	A	29	[2]	1'b0	RW		Change PCLK phase
dly_digi_PCLK[3:0]	A	28	[3:0]	4'b0000	RW		PCLK timing delay delay = dly_digi_PCLK*0.4 ns
d11_pad_en	A	2A	[7]	1'b0	RW		D11 pad control 1'b0 : disable 1'b1 : enable
d10_pad_en	A	2A	[6]	1'b0	RW		D10 pad control 1'b0 : disable 1'b1 : enable
d9_pad_en	A	2A	[5]	1'b0	RW		D9 pad control 1'b0 : disable 1'b1 : enable
d8_pad_en	A	2A	[4]	1'b0	RW		D8 pad control 1'b0 : disable 1'b1 : enable
d7_pad_en	A	2A	[3]	1'b0	RW		D7 pad control 1'b0 : disable 1'b1 : enable
d6_pad_en	A	2A	[2]	1'b0	RW		D6 pad control 1'b0 : disable 1'b1 : enable
d5_pad_en	A	2A	[1]	1'b0	RW		D5 pad control 1'b0 : disable 1'b1 : enable
d4_pad_en	A	2A	[0]	1'b0	RW		D4 pad control 1'b0 : disable 1'b1 : enable

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
d3_pad_en	A	2B	[7]	1'b0	RW		D3 pad control 1'b0 : disable 1'b1 : enable
d2_pad_en	A	2B	[6]	1'b0	RW		D2 pad control 1'b0 : disable 1'b1 : enable
d1_pad_en	A	2B	[5]	1'b0	RW		D1 pad control 1'b0 : disable 1'b1 : enable
d0_pad_en	A	2B	[4]	1'b0	RW		D0 pad control 1'b0 : disable 1'b1 : enable
pad_drv[1:0]	A	28	[7:6]	2'b00	RW		Data pad drivability control
dpad_swap	A	29	[1]	1'b0	RW		Data pad swap option 1'b0 : [MSB:LSB] 1'b1 : [LSB:MSB]
dly_d11_ctrl[2:0]	A	2C	[2:0]	3'b000	RW		D11 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d10_ctrl[2:0]	A	2D	[6:4]	3'b000	RW		D10 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d9_ctrl[2:0]	A	2D	[2:0]	3'b000	RW		D9 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d8_ctrl[2:0]	A	2E	[6:4]	3'b000	RW		D8 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d7_ctrl[2:0]	A	2E	[2:0]	3'b000	RW		D7 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d6_ctrl[2:0]	A	2F	[6:4]	3'b000	RW		D6 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d5_ctrl[2:0]	A	2F	[2:0]	3'b000	RW		D5 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d4_ctrl[2:0]	A	30	[6:4]	3'b000	RW		D4 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d3_ctrl[2:0]	A	30	[2:0]	3'b000	RW		D3 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d2_ctrl[2:0]	A	21	[6:4]	3'b000	RW		D2 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d1_ctrl[2:0]	A	21	[2:0]	3'b000	RW		D1 timing delay delay = dly_hsync_ctrl*0.8 ns
dly_d0_ctrl[2:0]	A	22	[6:4]	3'b000	RW		D0 timing delay delay = dly_hsync_ctrl*0.8 ns

Power-On/Off Sequence

The recommended power-up sequence for the PK3130K is shown in Figure 11. The available power supplies(HVDD, AVDD, RVDD, DVDD, DVDDM, DVDDPLL, RSTB) must have the separation specified below.

1. HVDD, AVDD, RVDD, DVDD, DVDDM and DVDDPLL don't matter if you power-on it at the same time.
2. RSTB must remain low(active low) for at least 8 clocks after power is stabilized.

The power-on/off sequence is illustrated in Table 9

Table 9 Power-on/off sequence

Symbol	Descriptions	Min	Typ	Max	Unit
t1	HVDD risiging to AVDD, RVDD, DVDD, DVDDM, DVDDPLL rising	0	-	100	ms
t2	Sensor reset timing	8	-	-	clock
t3	AVDD, RVDD, DVDD, DVDDM, DVDDPLL falling to HVDD falling	0	-	-	ms

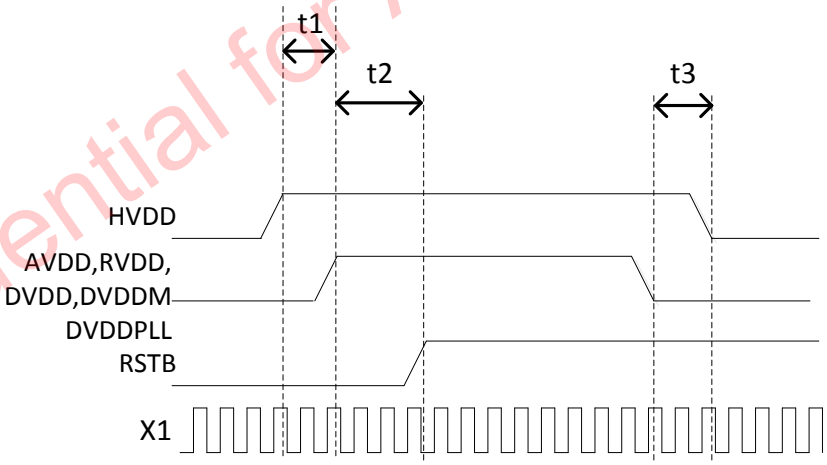


Figure 11 Timing diagram of power-on/off sequence

Clock

Figure 12 shows how the internal system clock(s) are generated in the PK3130K. System input crystal clock is processed in PLL block to generate higher frequency clock (vco clock). vco clock is further processed by divider circuit to generate necessary internal system clocks.

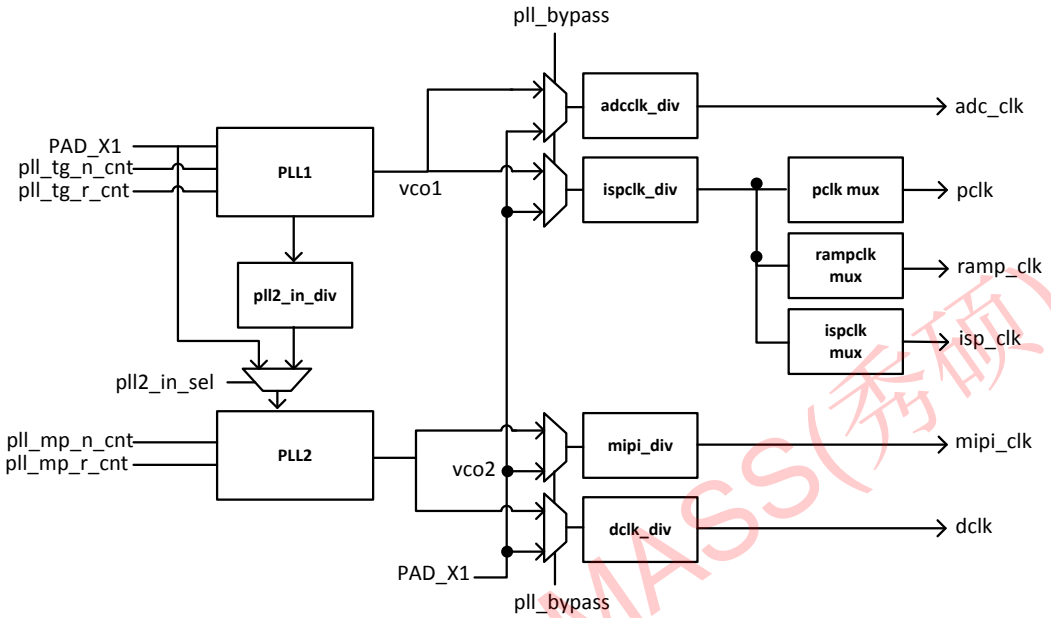


Figure 12 Internal system clock generation process

- PAD_X1 : PLL input clock
- vco1 : PLL1 output clock
- vco2 : PLL2 output clock
- pclk : Internal system clock for TG block system
- ramp_clk : Internal system clock for Ramp generaion
- isp_clk : Internal system clock for ISP block system
- dclk : Use parallel output PCLK or MIPI 1 byte data clock
- mipi_clk : MIPI serial output clock

PLL

Phase-locked loop (PLL) recieves input X1 clock as input reference frequency. X1 frequency is amplified to achieve higher frequency clock. Constraints and basic informations on PLL are as follows:

- Frequency of vco(PLL output clock) should be 148.5MHz ≤ vco1 ≤ 520MHz, 148.5MHz ≤ vco2 ≤ 520MHz.
- Vco1 = X1 × (pll_tg_n_cnt)/(pll_tg_r_cnt) ; (pll_tg_n_cnt) ≥ 12
- Vco2 = X1 × (pll_mp_n_cnt)/(pll_mp_r_cnt) ; (pll_mp_n_cnt) ≥ 12
- PLL Lock time should be greater than 200us.
- Initial PLL setting is disabled

Table 10 Register table - PLL

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
plltg_pd	A	48	[5]	1'b1	RW		PLL1 power down mode 1'b0 : pll1 power on 1'b1 : pll1 power down
pllmp_pd	A	48	[3]	1'b1	RW		PLL2 power down mode 1'b0 : pll2 power on 1'b1 : pll2 power down
pll_bypass	A	48	[4]	1'b1	RW		PLL bypass 1'b0 : use pll mode 1'b1 : pll bypass mode
pll_tg_n_cnt	A	4B	[7:0]	0x2C	RW		PLL multiplication factor for TG
pll_tg_r_cnt	A	4C	[4:0]	0x04	RW		PLL division factor for TG
pll_mp_n_cnt	A	4D	[7:0]	0x4D	RW		PLL multiplication factor for MIPI
pll_mp_r_cnt	A	4E	[4:0]	0x04	RW		PLL division factor for MIPI

Clock Divider

Clock divider block takes vco clock as input, and the input is divided by division factor to generate internal system clock(s).

Table 11 Register table - Clock divider

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
adcclk_div	A	25	[7:6]	2'b00	RW	aev	ADC clock divider isp_clk =vco1/(2^adcclk_div)
ispclk_div	A	25	[5:4]	2'b01	RW	aev	ISP clock divider isp_clk =vco1/(2^ispclk_div)
pll2_in_div	A	25	[3:2]	2'b10	RW	aev	PLL2 input clock divider isp_clk =isp_clk/(2^pll2_in_div)
pll2_in_sel	A	25	[1]	1'b1	RW	aev	PLL2 input clock selector 1'b0 : pll2_in_div output clock 1'b1 : main OSC clock
pclk_div	A	25	[0]	1'b1	RW	aev	pclk clock divider pclk = isp_clk/(2^pclk_div)
mipiclk_div	A	26	[7:6]	2'b00	RW	aev	MIPI clock divider mipi_clk =vco2/(2^mipiclk_div)
dclk_div	A	26	[5:3]	3'b010	RW	aev	MIPI byte clock divider dclk =vco2/(2^dclk_div)
digi_ispclk_div	A	26	[2]	1'b0	RW	aev	ISP clock divider isp_clk = isp_clk/(2^digi_ispclk_div)
rampclk_div	A	26	[1]	1'b0	RW	aev	Ramp clock divider ramp_clk = isp_clk/(2^rampclk_div)
clkoff	A	27	[3]	1'b0	RW		Clock pad kill enable 1'b0 : disable (not kill) 1'b1 : enable (kill)

PLL and Clock Setting Sequence

- When using PLL, set-up sequence, shown in Figure 13, is necessary.
- I2C update timing register, i2c_control_1, is changed before setting clock dividers to immediately apply clock divider settings (refer to I2C section).

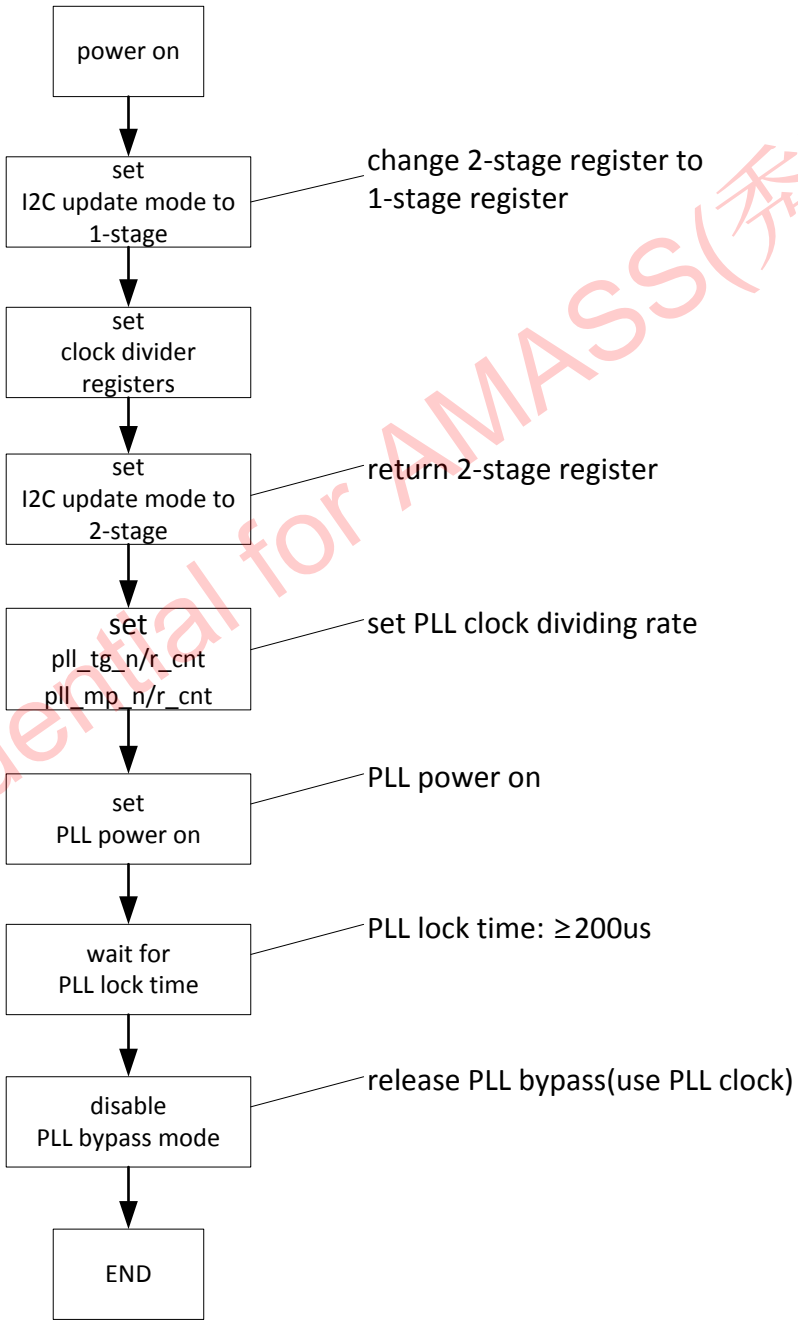


Figure 13 Clock setting sequence

Standby Mode

The PK3130K provide hardware standby. Hardware standby mode is controlled by STDBY PAD. I2C communication cannot be used while hardware standby mode is set.

Table 12 Register Table - STDBY mode

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
stdby_level[1:0]	A	27	[5:4]	2'b10	RW		Output data pad stdby level selector 2'b00 : low 2'b01 : high 2'b1x : hiz

System Reset

The PK3130K has two methods to reset: hard reset and soft reset hard reset signal from RSTB PAD must remain low (active low) for at least 8 master clocks to correctly reset the sensor. all registers are set to their default values after reset.
Figure 14 shows device soft reset by setting softreset register through I2C interface. When softreset register is set, async_rstb (asynchronous reset) and sync_rstb signal changes from 1 to 0 and holds for 1 clock of SCL. Afterward, async_rstb is set back to 1 while sync_rstb holds 0 for another 16 clocks of pclk for stable reset operation. Therefore, the PK3130K requires at least 1 clock of SCL and 16 clocks of pclk to perform a soft reset operation.

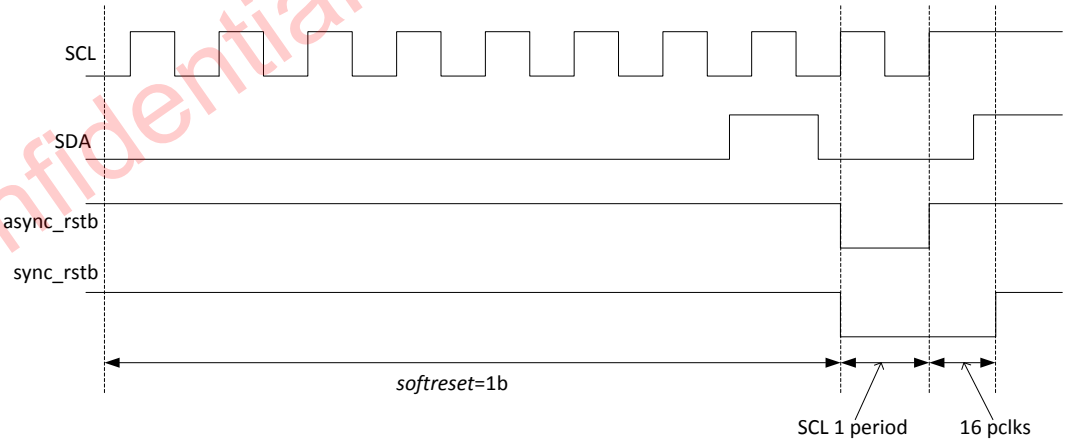


Figure 14 Soft reset

Table 13 Register Table - Soft reset

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
softreset	A	24	[0]	0x00	RW		Soft reset 1'b0 : disable 1'b1 : enable (after succesful reset value reverts to 0)

I2C Interface

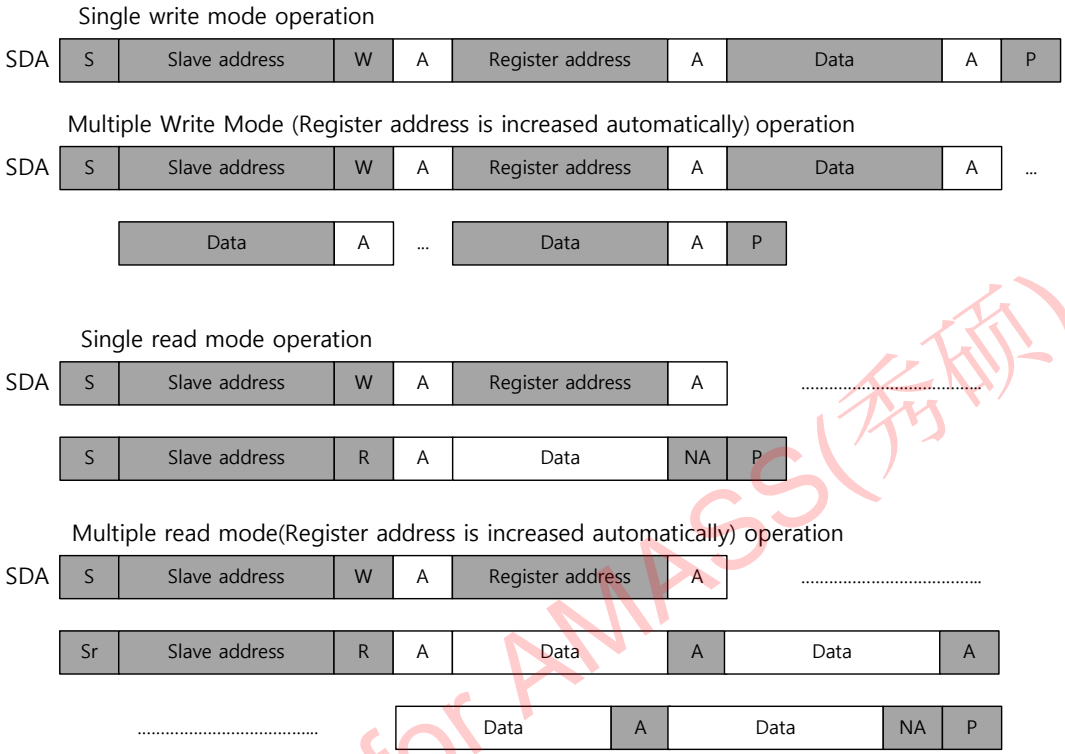
I2C Communication

I2C communication is a serial interface which utilizes SCL/SDA lines to transfer 8-bit data per transaction. The PK3130K includes only I2C slave function and requires external master to access the internal registers. Each transaction requires 8-bit data and 1-bit acknowledge bit. There are four types of operations supported in the PK3130K's I2C operation: single write, multiple write, single read, multiple read.

In single write operation, after the start state, 7-bit slave address and write bit are transmitted from master device to the PK3130K. If correct slave address is detected, the PK3130K reply with acknowledge bit as confirmation of valid address. Then master device transmits register address and waits for acknowledge bit from the PK3130K. Lastly, 8-bit data is sent to the PK3130K and waits for acknowledge bit again. Once acknowledge bit is recieved, master device announces the stop state to terminate I2C communication.

Multiple write operation works exactly the same until stop state procedure. Instead of announcing the stop state, master device transmits more data. If the PK3130K detects multiple write operation, any data stream following the first 8-bit is stored in subsequent register addresses of the first register address.

Read operation consists of two sub-procedure: address write and data read. The procedure is performed exactly same as register address write procedure from single write operation. Afterward, master device announces repeated start and transmit slave address with read bit. When the PK3130K detects read operation, the PK3130K sends acknowledge bit to master device, then reads register corresponding to register address. The PK3130K transmits read data to master device and waits for master device to respond. If master device responds with no acknowledge bit followed by stop state, read operation is terminated. On the other hand, if master device responds with acknowledge, the PK3130K reads the subsequent register and transmits again. As long as master device replies with acknowledge bit after each data transaction, the PK3130K will continuously read the subsequent register and transmit until no acknolwedge bit followed by stop state is presented. If only one 8-bit data is read, the procedure is single read operation. whereas, reading more than 8-bit data is multiple read operation. Figure 15 shows read/write operation of I2C communication.



Slave address can be extended 60h to 67h by CADDR0/1 PAD

CADDR1	CADDR0	write address	read address
1'b0	1'b0	60h	61h
1'b0	1'b1	62h	63h
1'b1	1'b0	64h	65h
1'b1	1'b1	66h	67h

R/W : Read/Write selection, High = read / Low = write
A : Acknowledge bit, NA : No Acknowledge, DATA : 8-bit data. P : Stop condition
S : Start condition, Sr : Repeated start(start without preceding stop)

Figure 15 I2C functional description

Register Update Timing

Registers has three different types of update timing: "aev" update, regular update, "m_wr" update, "wr_en" update. Registers with "aev" update type updates new values from I2C write operation at the last line of the frame. Whereas, registers with regular update type apply new values immediately after I2C write operation. However, By changing updatecontrol register value, register updates for "aev" type can either be disabled or be updated immediately. "m_wr" update registers only updates MPU genlock related registers. When "m_wr" is set, MPU genlock related registers value are updated and applied to the system in the next frame. "wr_en" update registers only applies to exposure related registers (integration time, global gain, digital gain). Due to exposure controls being split across several registers which leads to unreliable exposure updates if updated one

by one, all exposure related registers' update timing is control by reg_wr_en register. When reg_wr_en register is set, "wr_en" update registers are updated simultaneously.

Table 14 shows registers relevant to I2C update control.

Table 14 I2C update timing control

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
updatecontrol[3:0]	A	1A	[7:4]	4'b0101	RW		Control I2C register with autov, aev update type LSB 2-bit updatecontrol[1:0] controls aev update and MSB 2-bit updatecontrol[3:2] controls autov update 2'b00 : no update 2'b01 : aev update 2'b1x : immediate aev update
wr_en	B	BF	[0]	0x00	RW		Update exposure related register 1'b0 : no update 1'b1 : wr_en set
wr_en_off	B	B3	[0]	0x00	RW		Update exposure related register 1'b0 : update @ wr_en = 1'b1 1'b1 : Immediately update
mpu_gen_write	B	A9	[0]	0x0	RW		Genlock I2C register update

Initialization Timing for I2C Interface

Register control through I2C communication is possible at the point where 16 pclk has passed after RSTB becomes high (refer to Figure 16).

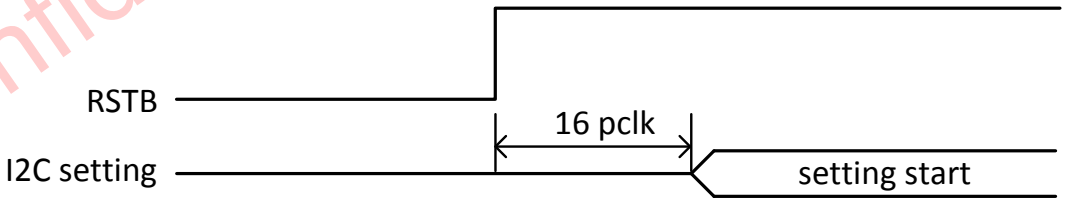


Figure 16 Available timting for I2C communication after system reset.

Exposure Control

Integration Time

The PK3130K employs rolling shutter [1] for capturing image. Reset operation initializes ROBP and active pixel region in sequence row by row. Readout process reads pixel data stored in photodetector at the identical order and speed as reset operation. The difference in time between reset and readout operation is known as integration time (refer to Figure 17). Integration time controls photodetector's level of exposure to light. Integration time can be adjusted in line unit level (line inttime) and column unit level (column inttime). Under the assumption of fixed frame structure, the maximum line inttime is "frame height - 5" and column inttime is "frame width - 1". Upper 16 bits of inttime register represent number of lines for line inttime and lower 8 bits of inttime register represent number of column inttime. The minimum exposure is determined by the minline_c-1 value. The final line exposure is determined by "inttime_h / m + minline_c".The minline_c should not be lower than *1*.

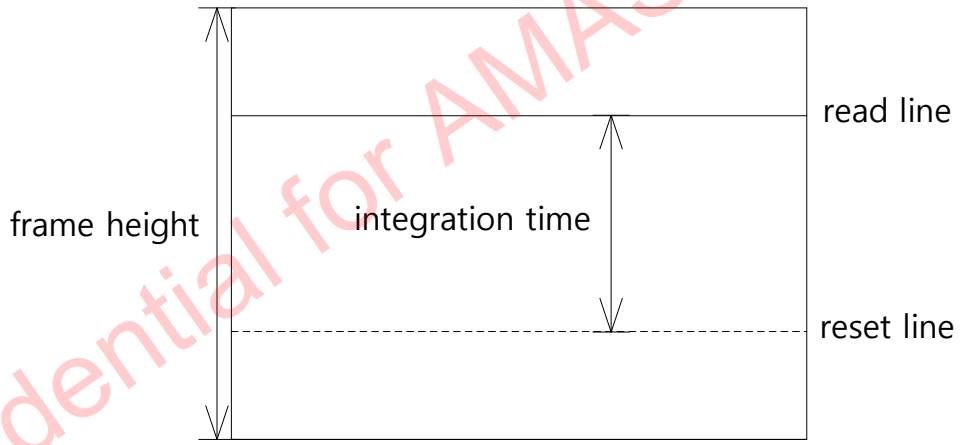


Figure 17 Fundamental concept of integration time

Table 15 shows registers relevant to integration time.

Table 15 Register Table - Integration time

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
inttime_h	B	C0	[7:0]	0x01	RW	wr_en	Integration time (line) High Byte
inttime_m	B	C1	[7:0]	0x40	RW	wr_en	Integration time (line) Low Byte
inttime_l	B	C2	[7:0]	0x00	RW	wr_en	Iinteration time (column)
frmvar_en	B	1B	[6]	1'b0	RW		Variable frame enable 1'b0 : disable 1'b1 : enable
minline_c	B	17	[1:0]	0x02	RW		Minimum expose

[1] Image capture method in which each frame is scanned row by row instead of capturing entire frame at once

Global Gain

Global gain affects analog gain level of comparators, which determines Bayer data values. In the PK3130K, global gain is ranged from 0x00 to 0x6F.

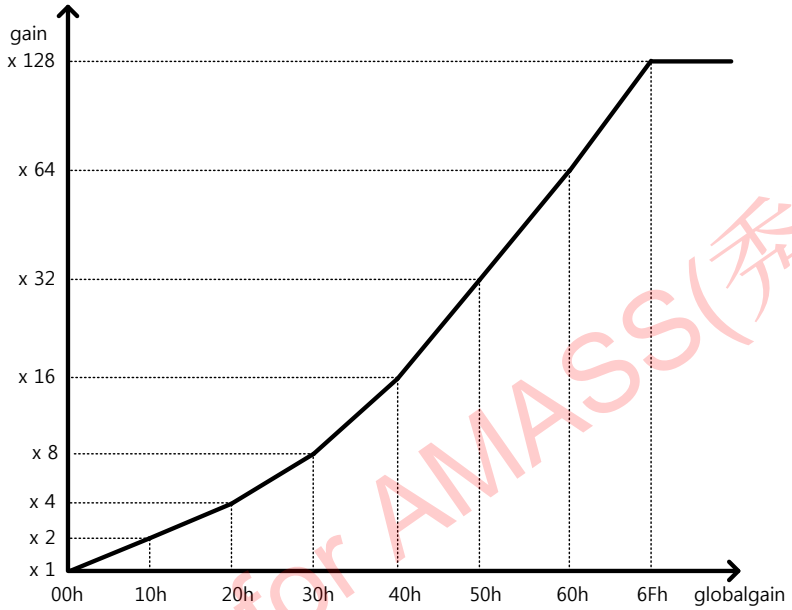


Figure 18 Globalgain's gain

Table 16 shows registers relevant to global gain.

Table 16 Register Table - Global gain

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
globalgain	B	C3	[7:0]	0x00	RW	wr_en	Analog gain
globalgain_max	B	D2	[7:0]	0x6F	RW		Maximum globla gain value

Digital Gain

Analog signal is converted to digital value through ADC operation, and the digital value can be amplified by digital gain. digitalgain register's upper 4 bits are positive integer and lower 4 bits are fraction.

Table 17 shows registers relevant to digital gain

Table 17 Register Table - Digital gain

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
digitalgain_l	B	C4	[7:0]	0x10	RW	wr_en	Digital gain of long data
digitalgain_s	B	C5	[7:0]	0x10	RW	wr_en	Digital gain of short data

Genlock

Genlock Configuration

Generator locking (genlock) synchronizes internal synchronous timing of master and slave. The PK3130K includes two different methods to achieve genlock. One is genlock sync method and the other is genlock I2C method.

- Genlock Sync Method
  Master device generates reference synchronous signal, which outputs via GENLOCK pad. Slave device receives the reference synchronous signal via its GENLOCK pad to achieve genlock. Figure 19 shows example of genlock sync set-up for using external MPU as master, whereas Figure 20 uses the PK3130K as master.
  - GENLOCK signal
    Slave device requires at least 4 pclk width of the reference synchronous signal for reliable genlock operation (refer to Figure 21). If the PK3130K is the master device, the signal width can be adjusted by changing genlock_width register.

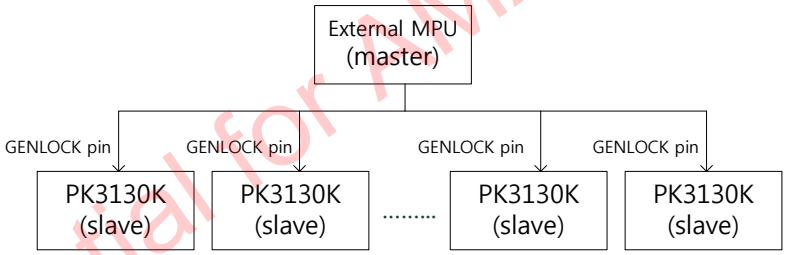


Figure 19 Genlock Sync configuration with external MPU

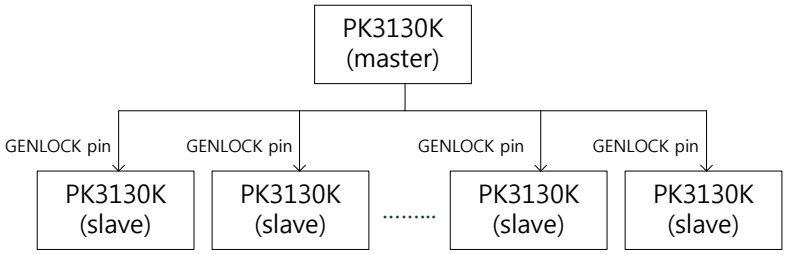


Figure 20 Genlock Sync configuration with another the PK3130K

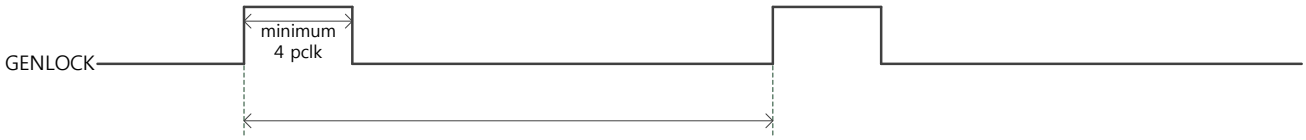


Figure 21 GENLOCK reference signal waveform

- Genlock I2C Method
  External MPU (master device) controls the slave devices' genlock registers to achieve genlock. Figure 22 shows example of genlock I2C set-up.

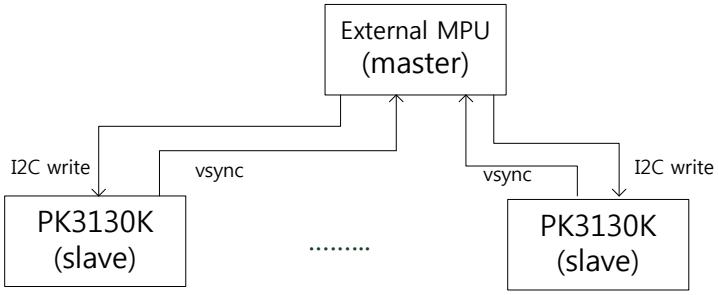


Figure 22 Genlock I2C configuration

Genlock Sync Method

- Genlock Master Mode (genlock_master = 1'b1)
  In the case of the PK3130K acting as the master device, if internal rcount and ccount value are equal to rcount_genlock and ccount_genlock respectively, master device outputs the reference synchronous signal via GENLOCK pad and the signal remains high for genlock_width * pclk.

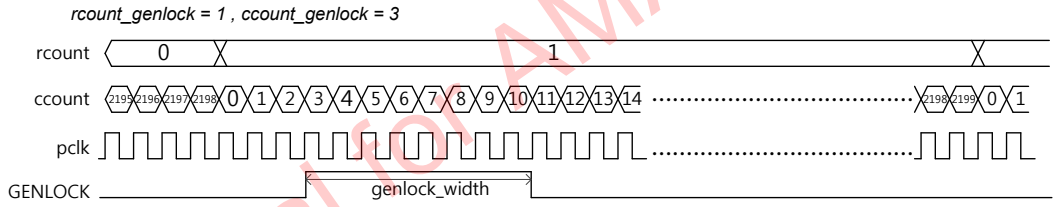


Figure 23 Example of timing diagram @master mode

- Genlock Slave Mode (genlock_master = 1'b0)
  In the case of the PK3130K acting as the slave device, if the slave device receives reference synchronous signal from master device via GENLOCK pad, internal rcount and ccount are initialized to rcount_genlock and (ccount_genlock + 1) respectively at rising edge of the signal.

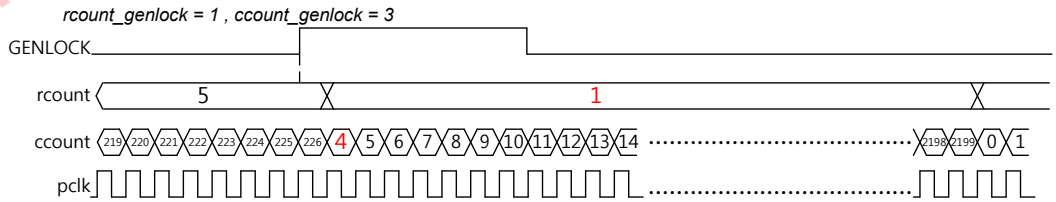


Figure 24 Example of timing diagram @slave mode

Note If the synchronization timing difference between master and slave is large, sudden shift in brightness may occur in slave device's image.

Figure 25 shows timing diagram of genlock sync method with the PK3130K as master device.

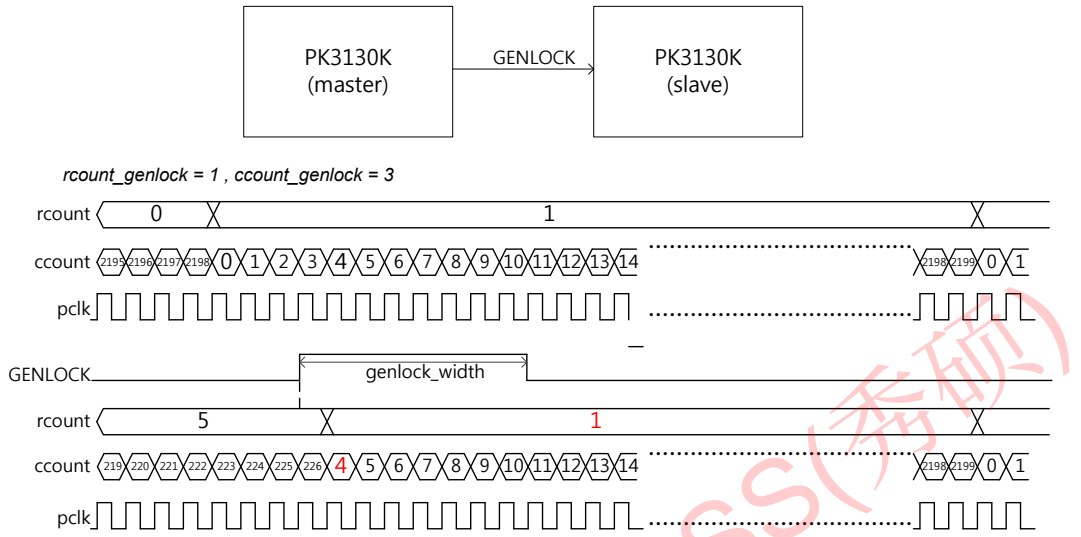


Figure 25 Example of timing diagram @genlock sync mode

Table 18 shows registers relevant to genlock sync method.

Table 18 Register Table - genlock sync mode

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
genlock_pad_en	A	29	[0]	1'b0	RW		GENLOCK pad enable 1'b0 : disable 1'b1 : enable
genlock_en	B	1B	[5]	1'b0	RW		GENLOCK enable 1'b0 : disable 1'b1 : enable
genlock_master	B	1B	[4]	1'b0	RW		GENLOCK master 1'b0 : slave 1'b1 : master
rcount_genlock_h	B	4B	[4:0]	0x00	RW		Genlock row count High Byte
rcount_genlock_l	B	4C	[7:0]	0x01	RW		Genlock row count Low Byte
ccount_genlock_h	B	4D	[4:0]	0x00	RW		Genlock column count High Byte
ccount_genlock_l	B	4E	[7:0]	0x01	RW		Genlock column count Low Byte
genlock_width	B	16	[7:0]	0x10	RW		Genlock puslse width

Genlock I2C Method

Generic set-up for genlock I2C method consists of one external MPU (master device) and at least two the PK3130K (slave device). External MPU monitors slave devices' vsync signals and evaluate synchronization differences. Afterward, external MPU controls genlock settings, based on the evaluation result, through I2C communication to achieve genlock state. mpu_gen_line and mpu_gen_col registers controls synchronization timing in frame unit and column unit respectively. The range of mpu_gen_line is from -3 to frameheight and mpu_gen_col is from (framewidth-1200) to framewidth.

Figure 26 shows timing diagram of genlock I2C method example. genlock compensation of +A value is applied in the following frame as shown in Figure 26. Genlock related registers have to retain the same value in the

frame where genlock compensation is expected, because mpu_gen_line and mpu_gen_col registers are updated before the genlock compensation can occur.

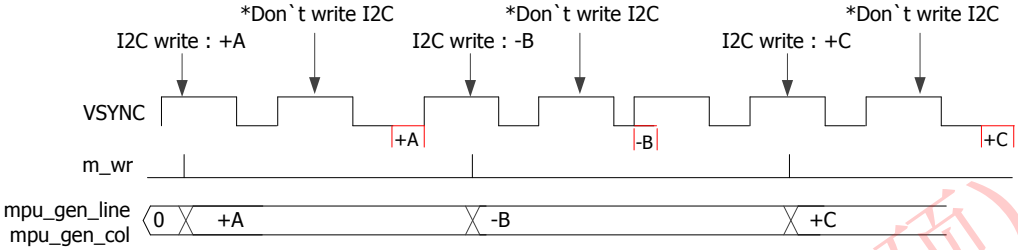


Figure 26 Example of timing diagram @genlock I2C mode

Table 19 shows registers relevant to genlock I2C method.

Table 19 Register Table - Genlock I2C mode

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mpu_gen_line_h	B	A5	[4:0]	0x00	RW	m_wr	Frame height resized by mpu_gen_line High Byte
mpu_gen_line_l	B	A6	[7:0]	0x00	RW	m_wr	Frame height resized by mpu_gen_line Low Byte
mpu_gen_col_h	B	A7	[4:0]	0x00	RW	m_wr	Frame width resized by mpu_gen_col High Byte
mpu_gen_col_l	B	A8	[7:0]	0x00	RW	m_wr	Frame width resized by mpu_gen_col Low Byte
mpu_gen_write	B	A9	[0]	0x0	RW		Genlock I2C register update

Test Pattern Control

TP control generates test images. Test images type can be selected by setting tp_control_0 registers. Some test pattern colors and color intensity can be modified by tp_control_1~5 registers. In addition, tp_seq register controls sequence of R, G1, G2, B for test pattern.

- R : {tp_control_1, tp_control_5[7:6]}
- G1 : {tp_control_2, tp_control_5[5:4]}
- G2 : {tp_control_3, tp_control_5[3:2]}
- B : {tp_control_4, tp_control_5[1:0]}

Table 20 shows registers relevant to Test Pattern Control

Table 20 Register Table - Test pattern

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
tp_control_0	B	B4	[7:0]	0x00	RW		Test pattern selection

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
tp_control_1	B	B5	[7:0]	0x00	RW		R[9:2] color for test pattern
tp_control_2	B	B6	[7:0]	0x00	RW		G1[9:2] color for test pattern
tp_control_3	B	B7	[7:0]	0x00	RW		G2[9:2] color for test pattern
tp_control_4	B	B8	[7:0]	0x00	RW		B[9:2] color for test pattern
tp_control_5	B	B9	[7:0]	0x00	RW		{R[1:0],G1[1:0],G2[1:0],B[1:0]} for test pattern
tp_width_h	B	BA	[2:0]	0x05	RW		Test pattern width for color bar pattern High Byte
tp_width_l	B	BB	[7:0]	0x10	RW		Test pattern width for color bar pattern Low Byte
tp_seq[1:0]	B	21	[3:2]	2'b00	RW		Test pattern block input sequence selection

Figure 27 shows test pattern images based on tp_control_0 value (tp_control_1~5 = FFh).

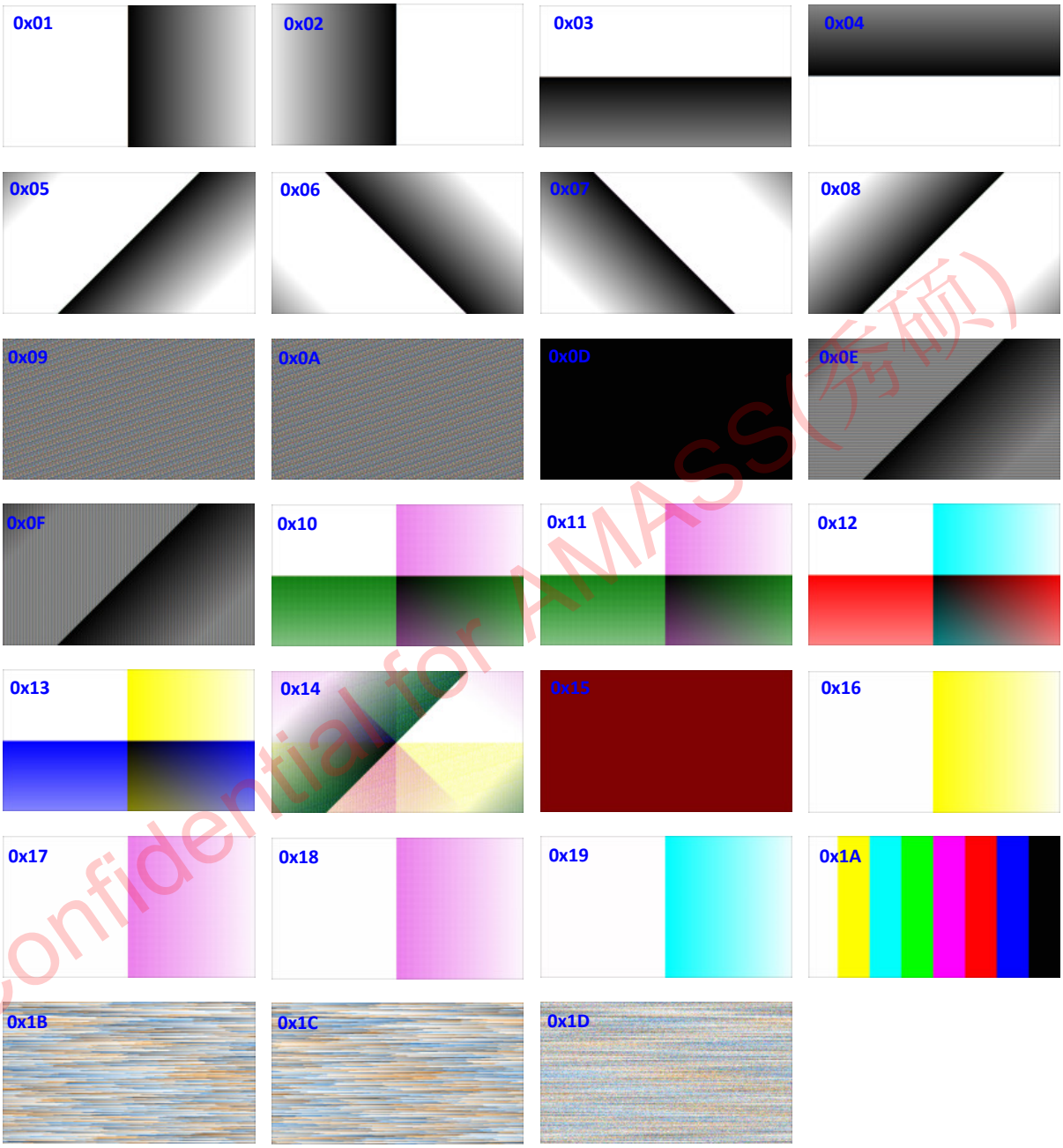


Figure 27 Test Pattern

MIPI

Reference For Design

MIPI design in the PK3130K is based on “MIPI Alliance Standard for Camera Serial Interface 2 (CSI-2), Version 1.00” and “MIPI Alliance Standard for D-PHY, Version 0.65” specification documents. Output of MIPI consists of one clock lane and two data lanes.

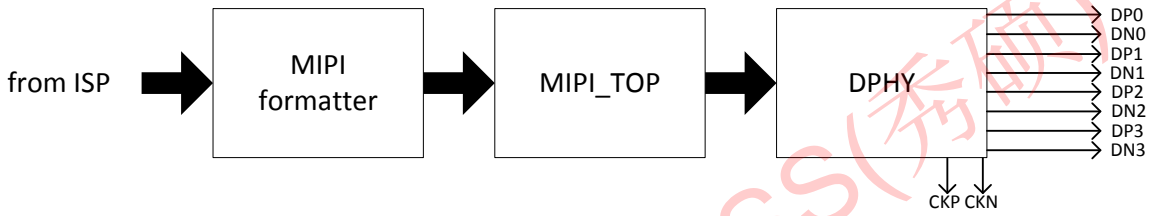


Figure 28 MIPI block diagram

MIPI Clock Relations

Figure 29 shows Clock and MIPI operation. The mipi_clk operates at a frequency four times faster than dclk. In addition, it is used for generating MIPI data and clock lane signal. The dclk is clock rate determined by isp_clk, raw bit, MIPI lane. Table 21 shows clock rate setting.

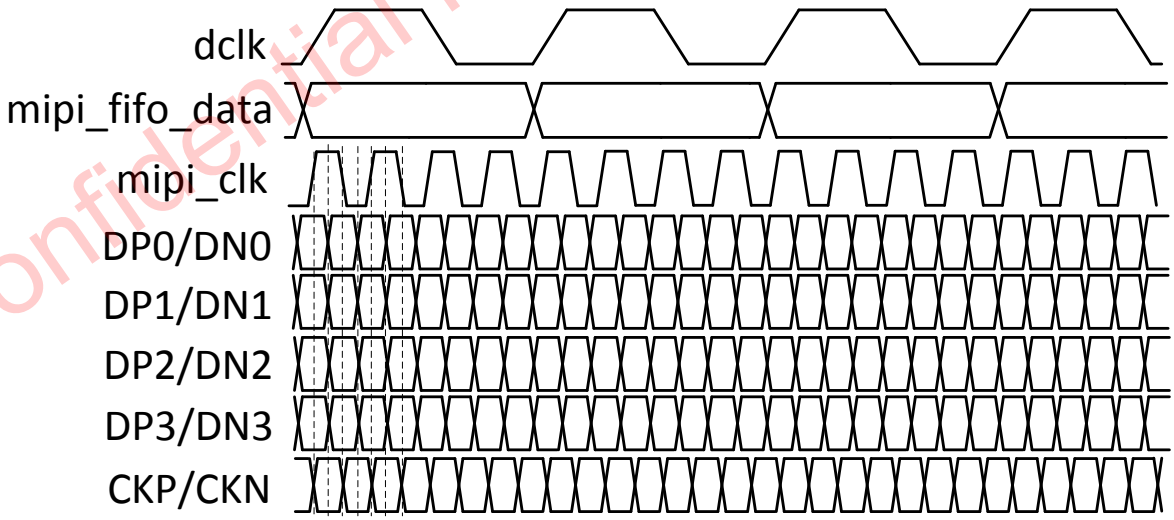


Figure 29 MIPI clock relations diagram

Table 21 MIPI clock with lane and frame rate

FPS	clock domain	MIPI 2 lane	MIPI 4 lane
		raw 12-bit(Mhz)	raw 12-bit(Mhz)
30fps	pclk	74.25	74.25
	isp_clk	148.5	148.5
	dclk	111.375	55.6875
	mipi_clk	445.5	222.75
	PLL2	445.5	445.5
	PLL1	297	297

Figure 30 shows raw 12-bit format.

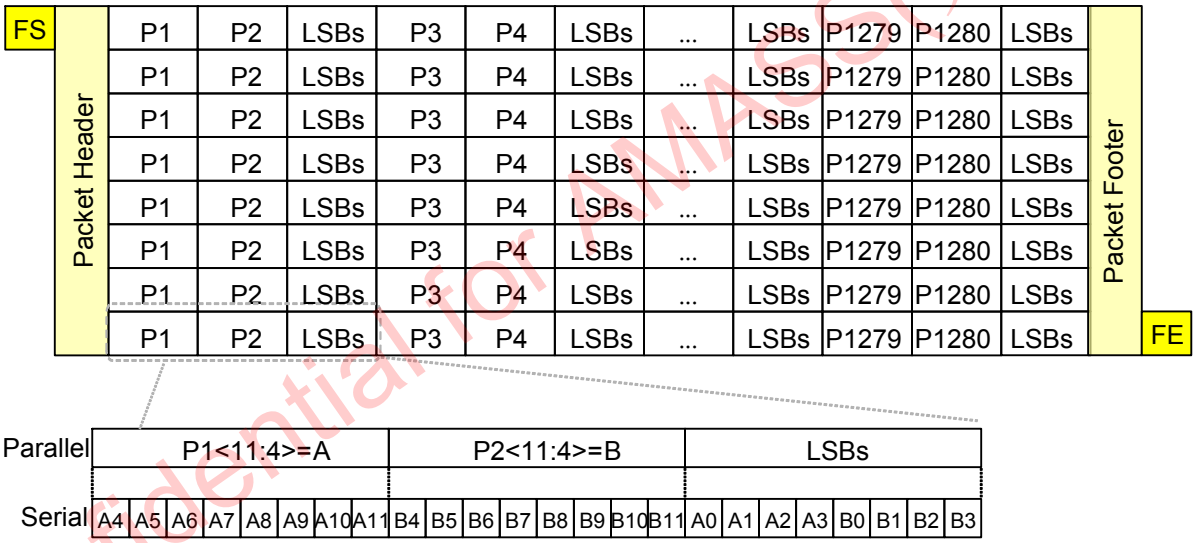


Figure 30 MIPI raw12 frame format

Table 22 shows registers relevant to MIPI and LVDS mode setting.

Table 22 Register Table - MIPI and LVDS mode setting

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mipi_en	F	04	[6]	1'b0	RW	aev	MIPI enable 1'b0 : disable 1'b1 : enable
clk_hs_mode	F	04	[4]	1'b1	RW	aev	MIPI clock lane hs mode 1'b0 : LP & HS mode 1b'1 : only HS mode
mipi_dbit[1:0]	F	08	[5:4]	2'b00	RW		Data seletor of the mipi_format block 2'b01 : mipi data 12 bit
mipi_lane	F	08	[3:2]	2'b00	RW		MIPI lane selector 2'b01 : 2 lane mode 2'b10 : 4 lane mode

MIPI Global Operation

- Data Unit and Bit Transmission Order

MIPI transmitter serially sends data in byte unit starting from LSB. Figure 31 shows MIPI data transfer order.

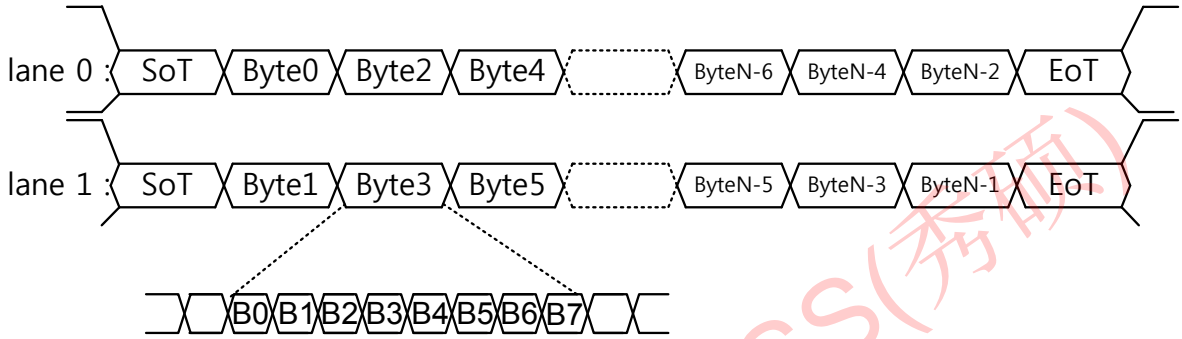


Figure 31 MIPI transmission order

- Lane States and Line Levels

Transmitter functions determine the Lane state by driving certain Line levels. During normal operation either a HS-TX or a LP-TX is driving a Lane. A HS-TX always drives the Lane differentially. The two LP-TX's drive the two Lines of a Lane independently and single-ended. This results in two possible High-Speed Lane states and four possible Low-Power Lane states. The High-Speed Lane states are Differential-0 and Differential-1. Low Power signaling is used for both Control mode and Escape mode. The interpretation of Low-Power Lane states depends on the mode of operation.

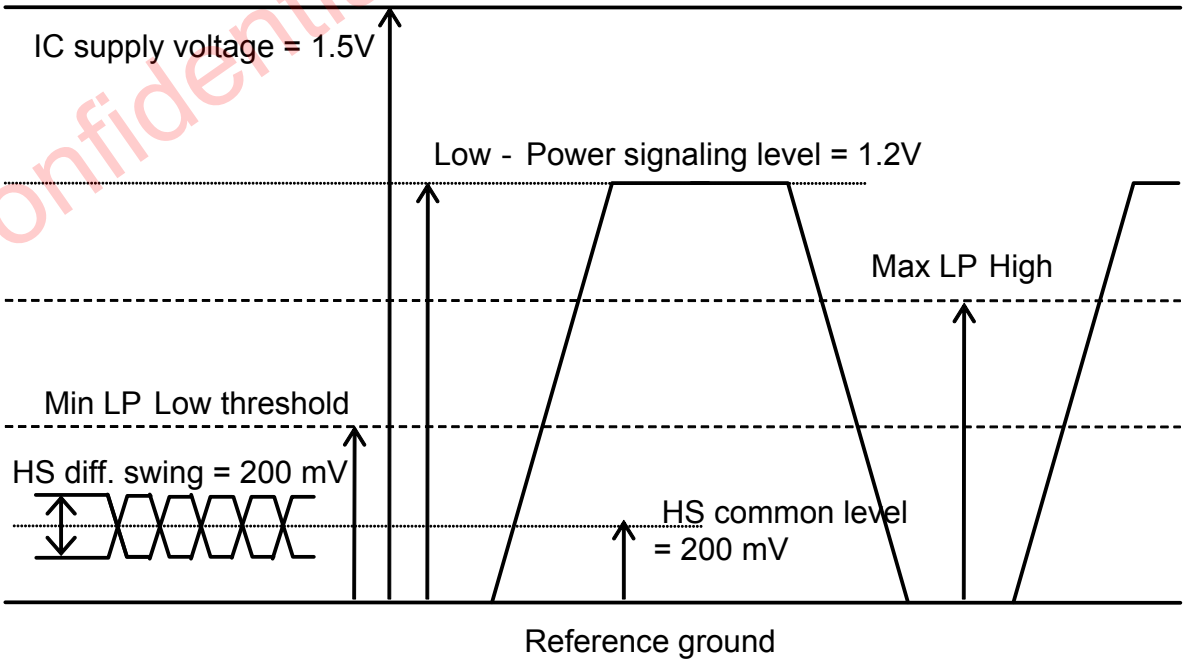


Figure 32 MIPI PAD levels

- Operating Modes : Control, High-Speed, and Escape

During normal operation a Data Lane will be either in Control or High-Speed mode. High-Speed Data transmission happens in bursts and starts from and ends at a Stop state (LP-11), which is by definition in Control mode. The Lane is only in high-speed mode during Data bursts. The sequence to enter high speed mode is : LP-11, LP-01, LP-00 at which point the Data Lane remains in high speed mode until a LP-11 is received. The special Escape mode can only be entered via a request within Control mode. The Data Lane shall always exit Escape mode and return to Control mode after detection of a Stop state. If not in High-Speed or Escape mode the Data Lane shall stay in Control mode. For Data Lanes and for Clock Lanes the Stop state serves as general standby state and may last for any period of time > T_{LPX}. Possible events starting from and ending in the Stop state are High-Speed Data Transmission burst (LP-11, LP-01, LP-00) and Escape mode request (LP-11, LP-10, LP-00, LP-01, LP-00). The Lane shall stay in the Stop state as long as no other state is presented on the Lane.

Table 23 Lane states description

State Code	Line Voltage Levels		High-Speed	Low-Power	
	DP-Line	DN-Line	Burst Mode	Control Mode	Escape Mode
HS-0	HS Low	HS High	Differential-0	N/A	N/A
HS-1	HS High	HS Low	Differential-1	N/A	N/A
LP-00	LP Low	LP Low	N/A	Bridge	Space
LP-01	LP Low	LP High	N/A	HS-Rqst	Mark-0
LP-10	LP High	LP Low	N/A	LP-Rqst	Mark-1
LP-11	LP High	LP High	N/A	Stop	N/A

Table 24 shows registers relevant to PAD level control based on MIPI state.

Table 24 Register Table - MIPI PAD control

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mipi_ck_control[3:0]	F	05	[7:4]	4'b1010	RW		MIPI CKP/CKN pad state control 4'b0000 : Normal operation mode 4'b0001 : CKP/CKN = LP-00 state 4'b0010 : CKP/CKN = LP-01 state 4'b0011 : CKP/CKN = LP-10 state 4'b0100 : CKP/CKN = LP-11 state 4'b0101 : CKP/CKN = HS-0 state 4'b0110 : CKP/CKN = HS-1 state 4'b0111 : CKP/CKN = Hi-z state 4'b1000 : CKP/CKN = ULP state 4'b1010 : CKP/CKN = power down
mipi_d0_control[3:0]	F	06	[7:4]	4'b1010	RW		MIPI DP0/DN0 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP0/DN0 = LP-00 state 4'b0010 : DP0/DN0 = LP-01 state 4'b0011 : DP0/DN0 = LP-10 state 4'b0100 : DP0/DN0 = LP-11 state 4'b0101 : DP0/DN0 = HS-0 state 4'b0110 : DP0/DN0 = HS-1 state 4'b0111 : DP0/DN0 = Hi-z state 4'b1000 : DP0/DN0 = ULP state 4'b1001 : DP0/DN0 = Serializer Test mode 4'b1010 : DP0/DN0 = power down

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mipi_d1_control[3:0]	F	06	[3:0]	4'b1010	RW		MIPI DP1/DN1 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP1/DN1 = LP-00 state 4'b0010 : DP1/DN1 = LP-01 state 4'b0011 : DP1/DN1 = LP-10 state 4'b0100 : DP1/DN1 = LP-11 state 4'b0101 : DP1/DN1 = HS-0 state 4'b0110 : DP1/DN1 = HS-1 state 4'b0111 : DP1/DN1 = Hi-z state 4'b1000 : DP1/DN1 = ULP state 4'b1001 : DP1/DN1 = Serializer Test mode 4'b1010 : DP1/DN1 = power down
mipi_d2_control[3:0]	F	07	[7:4]	4'b1010	RW		MIPI DP2/DN2 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP2/DN2 = LP-00 state 4'b0010 : DP2/DN2 = LP-01 state 4'b0011 : DP2/DN2 = LP-10 state 4'b0100 : DP2/DN2 = LP-11 state 4'b0101 : DP2/DN2 = HS-0 state 4'b0110 : DP2/DN2 = HS-1 state 4'b0111 : DP2/DN2 = Hi-z state 4'b1000 : DP2/DN2 = ULP state 4'b1001 : DP2/DN2 = Serializer Test mode 4'b1010 : DP2/DN2 = power down
mipi_d3_control[3:0]	F	07	[3:0]	4'b1010	RW		MIPI DP3/DN3 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP3/DN3 = LP-00 state 4'b0010 : DP3/DN3 = LP-01 state 4'b0011 : DP3/DN3 = LP-10 state 4'b0100 : DP3/DN3 = LP-11 state 4'b0101 : DP3/DN3 = HS-0 state 4'b0110 : DP3/DN3 = HS-1 state 4'b0111 : DP3/DN3 = Hi-z state 4'b1000 : DP3/DN3 = ULP state 4'b1001 : DP3/DN3 = Serializer Test mode 4'b1010 : DP3/DN3 = power down
d0_lane_swap	F	09	[7:6]	2'b00	RW		Data0 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
d1_lane_swap	F	09	[5:4]	2'b01	RW		Data1 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
d2_lane_swap	F	09	[3:2]	2'b10	RW		Data2 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							2'b10 : d2 lane data 2'b11 : d3 lane data
d3_lane_swap	F	09	[1:0]	2'b11	RW		Data3 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
mipi_ck_NP_swap	F	0B	[3]	1'b0	RW		Mipi clock N/P swap
d0_NP_swap	F	0B	[7]	1'b0	RW		Data0 N/P swap
d1_NP_swap	F	0B	[6]	1'b0	RW		Data1 N/P swap
d2_NP_swap	F	0B	[5]	1'b0	RW		Data2 N/P swap
d3_NP_swap	F	0B	[4]	1'b0	RW		Data3 N/P swap
mipi_test_d0	F	2D	[7:0]	0xAA	RW		MIPI test data 0 for HS state
mipi_test_d1	F	2E	[7:0]	0xFF	RW		MIPI test data 1 for HS state
mipi_test_d2	F	2F	[7:0]	0x00	RW		MIPI test data 2 for HS state
mipi_test_d3	F	30	[7:0]	0x55	RW		MIPI test data 3 for HS state

Low Level Protocol

The Low Level Protocol (LLP) is a byte oriented, packet based protocol that supports the transport of image data using Short and Long packet formats. For each packet structure, exit from the low power state followed by the Start of Transmission (SoT) sequence indicates the start of the packet. The End of Transmission (EoT) sequence followed by the low power state indicates the end of the packet.

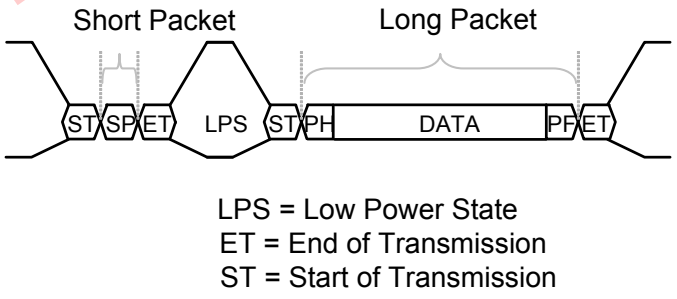


Figure 33 MIPI low level protocol

- Short Packet Format
  The PK3130K supports two types of Short Packets for frame synchronization : Frame Start (FS) Packet and Frame End (FE) Packet. Data ID field is 00h for FS and 01h for FE. Each image frame shall begin with a FS packet containing the Frame Start Code. The FS Packet shall be followed by one or more long packets containing image data. Each image frame shall end with a FE packet containing the Frame End Code. For FS and FE synchronization packets the Short Packet Data Field shall contain a 16-bit frame number. This frame number is the same for the FS and FE synchronization packets corresponding to a given frame. The 16-bit frame number shall always be non-zero to distinguish it from the use-case where frame number is inoperative and remains set to zero. The Error Correction Code (ECC) byte allows single-bit errors to be corrected and 2-bit errors to be detected in the Short Packet.

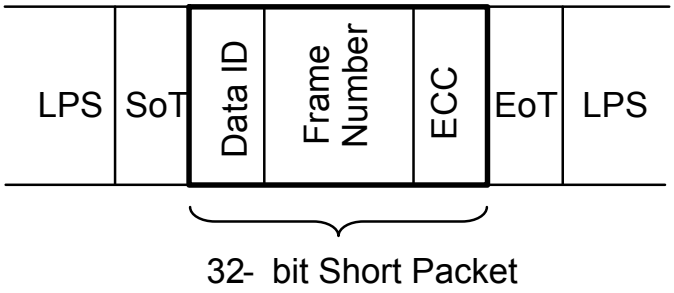


Figure 34 MIPI short packet structure

- Long Packet Format
  A Long Packet shall consist of 3 elements : a 32-bit Packet Header (PH), an application Data Payload with a variable number of 8-bit words and a 16-bit Packet Footer (PF). The Packet Header is further composed of three elements : an 8-bit Data Identifier, a 16-bit Word Count field and an 8-bit ECC. The Packet Footer has one element : a 16-bit checksum. The Word Count defines the number of 8-bit data words in the Data Payload between the end of the Packet Header and the start of the Packet Footer. Neither the Packet Header nor the Packet Footer shall be included in the Word Count. After the end of the Packet Header the receiver reads the next WC*8-bits data words of the Data Payload. While reading the Data Payload the receiver shall not look for any embedded sync codes. Therefore, there are no limitations on the value of a data word. Once the receiver has read the Data Payload it reads the checksum in the Packet Footer. In the generic case, the length of the Data Payload shall be a multiple of 8-bit data words. In addition, each image data format may impose additional restrictions on the length of the payload data. Each byte shall be transmitted least significant bit first. Multi-byte elements such as Word Count, Checksum and the Short packet 16-bit Data Field shall be transmitted least significant byte first.

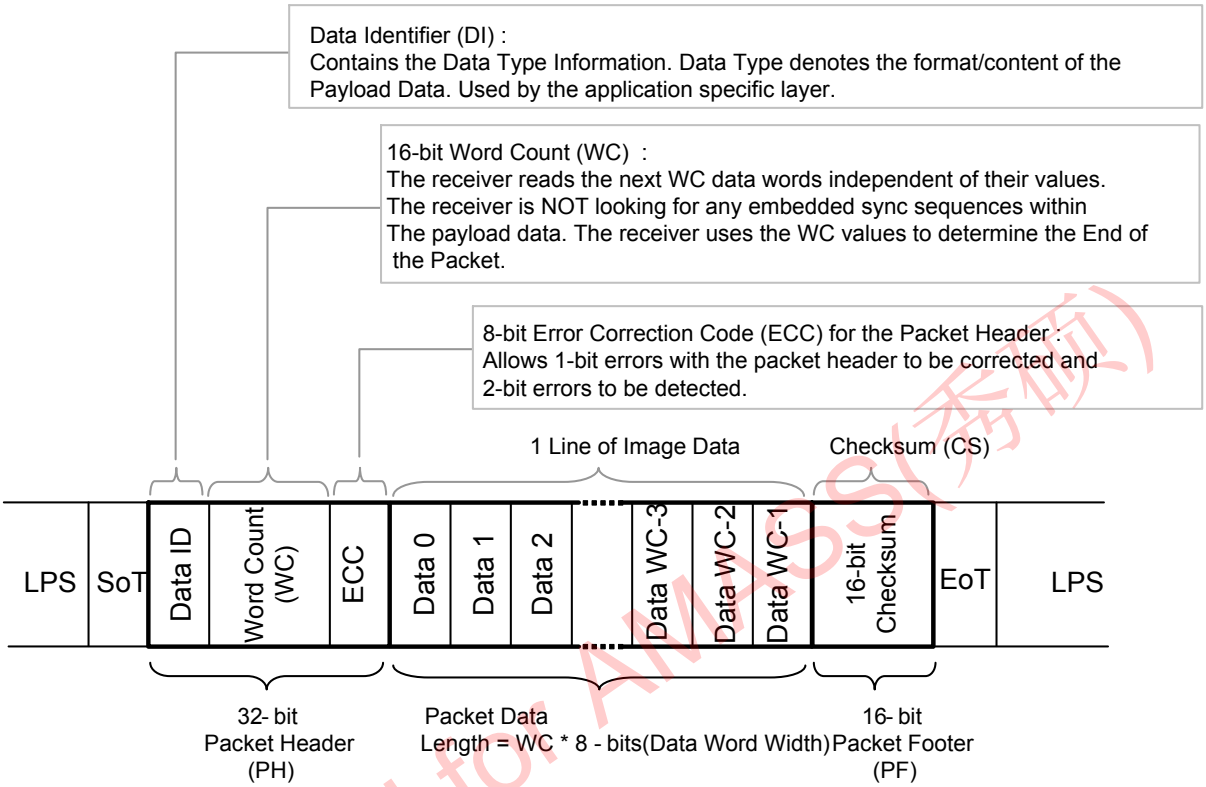


Figure 35 MIPI long packet structure

Table 25 MIPI Data Type and Data ID

Data ID	Data Type (Image Format)	Packet Type
00 hex	Frame start	Short
01 hex	Frame end	Short
2C hex	Raw bayer 12-bit	Long

Table 26 shows registers relevant to MIPI packet.

Table 26 Register Table - MIPI packet control

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mipi_pkt_size0_h	F	36	[7:0]	0x09	RW		MIPI word counter size 0 control for image data High Byte
mipi_pkt_size0_l	F	37	[7:0]	0x6F	RW		MIPI word counter size 0 control for image data Low Byte
mipi_data_id0	F	42	[7:0]	0x2B	RW		MIPI data 0 identifier

- Packet Spacing and Frame Format
  Between Low Level Protocol packets there must always be a transition into and out of the Low Power State (LPS). The packet spacing does not have to be a multiple of 8-bit data words as the receiver will synchronize to the correct byte boundary during the SoT sequence prior to the Packet Header of the next packet.

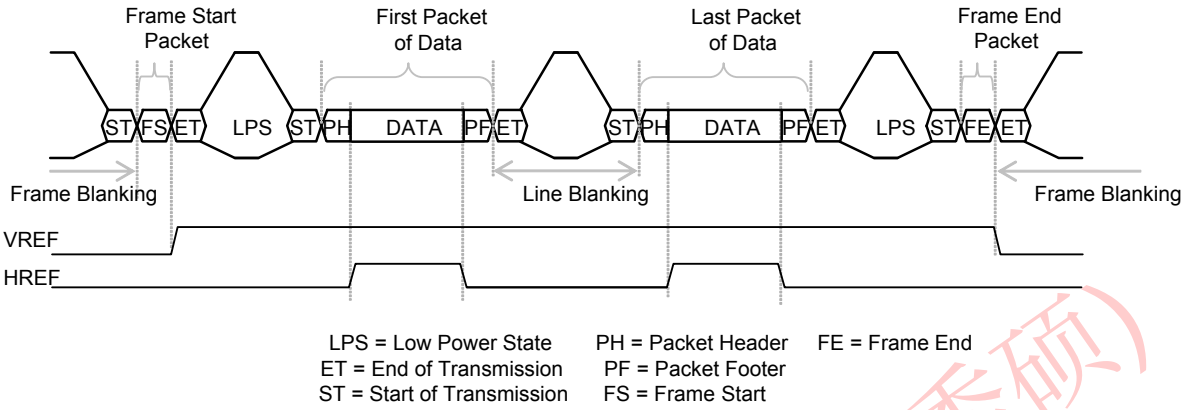


Figure 36 MIPI multiple packet

MIPI Electrical Characteristics

- Low-Power Transmitter
  The Low-Power transmitter is a slew-rate controlled push-pull driver. It is used for driving the Lines in all Low-Power operating modes. It is therefore important that the static power consumption of a LP transmitter be as low as possible. The slew-rate of signal transitions is bounded in order to keep EMI low.

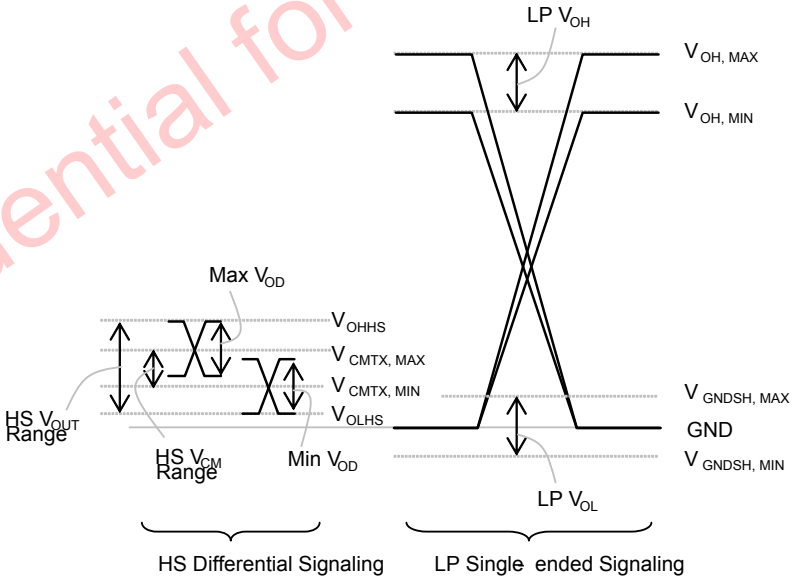


Figure 37 D-PHY signaling levels

Table 27 LP transmitter DC characteristics

Parameter	Description	Min	Nom	Max	Units	Notes
V_{OL}	Thevenin output low level	-50		50	mV	
V_{OH}	Thevenin output high level	1.1	1.2	1.3	V	
Z_{OLP}	Output Impedance of LP transmitter	110			Ohm	

Table 28 LP transmitter DC characteristics

Parameter	Description	Min	Nom	Max	Units	Notes
t_{RLP} / t_{FLP}	15%-85% rise time and fall time			25	ns	1, 5
t_{REOT}	30%-85% rise time in EOT state			35	ns	4, 5, 6
dV/dt_{SR}	Slew rate			120	mV/ns	1, 2, 3
C_{LOAD}	Load Capacitance	0		70	pF	

Notes 1. When the output is loaded with a capacitive load C_{LOAD}
Notes 2. When the output voltage is between 15% and below 85% of the fully settled LP signal levels.
Notes 3. Measured as average across 50mV segment of the output signal transition.
Notes 4. The rise-time of t_{REOT} starts from the HS common level at the moment the differential amplitude drops below 70mV, due to stopping the differential drive.
Notes 5. For capacitive loads from 0-70pF
Notes 6. With an additional load capacitance CCM between 0-60pF on the termination center tap at RX side of the link.

- High-Speed Transmitter
  The reference characteristic impedance level is 100 Ohm differential, 50 Ohm single-ended per wire, and 25 Ohm common-mode for both wires together. A HS differential signal driven on the DP and DN pins is generated by a differential output driver. For reference, DP is considered as the positive side and DN as the negative side. The Lane state is called Differential-1 (HS-1) when the potential on DP is higher than the potential of DN. The Lane state is called Differential-0 (HS-0), when the potential on DP is lower than the potential of DN.

The differential output voltage V_{OD} is defined as the difference of the voltages V_{DP} and V_{DN} at the DP and DN pins, respectively.

$V_{OD}=V_{DP}-V_{DN}$

The output voltages V_{DP} and V_{DN} at the DP and DN pins shall not exceed the high-speed output high voltage V_{OHHS}. The common-mode voltage V_{CMTX} is defined as the arithmetic mean value of the voltages at the DP and DN pins :

$V_{CMTX}=(V_{DP}+V_{DN})/2$

Table 29 HS transmitter DC characteristics

Parameter	Description	Min	Nom	Max	Units	Notes
V_{OD}	HS transmit differential voltage	140	200	270	mV	1
V_{CMTX}	HS transmit static common mode voltage	150	200	250	mV	1
$\triangle V_{OD}$	VOD mismatch when output is Differential-1 or Differne-tial-0			10	mV	
$\triangle V_{CMTX}$	VCMTX mismatch when output is Differential-1 or Differ-ential-0			5	mV	
V_{OHHS}	HS output high voltage			360	mV	1
Z_{OS}	Single ended output impedance	40	50	62.5	W	
$\triangle Z_{OS}$	Single ended output impedance mismatch			10	%	

Notes 1. Value when driving into load impedance

Table 30 HS transmitter AC characteristics

Parameter	Description	Min	Nom	Max	Units	Notes
$\Delta V_{CMTX(HF)}$	Common-level variation above 450MHz			15	mV_{RMS}	

Parameter	Description	Min	Nom	Max	Units	Notes
$\Delta V_{CMTX(LF)}$	Common-level variation between 50-450MHz			25	mV_{PEAK}	1
t_R / t_F	20%-80% rise time and fall time	150		$0.3UI_{NOM}$	ps	2
Notes 1. VPP is the voltage difference compared to the DC average common-mode potential. Notes 2. UI_{NOM} is the long term average Unit Interval.						

- High-Speed Data-Clock Timing
  The Master side of the Link shall send a differential clock to the Slave side to be used for data sampling. This clock is at a fixed nominal frequency and stable for the entire duration of a data transfer. The DDR [Double Data Rate] Clock signal maintains a quadrature phase relationship to the data signal. Data will be sampled by both the rising and falling edges of the Clock signal. The Clock signal is a differential signal. Use of the term “rising-edge” means “rising edge of the signal (CLPp – CLKn)” and similarly for “falling edge”. Therefore, the frequency of the Clock signal will be half the desired data rate in bits per second. The timing relationship of the DDR Clock differential signal to the NRZ Data differential signal is shown in Figure 38 . Data is launched in a quadrature relationship to the clock such that the Clock signal edge may be used directly by the receiver to sample the received data. The rising edge of the DDR Clock is sent during the first bit of each byte, such that the receiver can sample the bits of each byte starting with a rising edge.

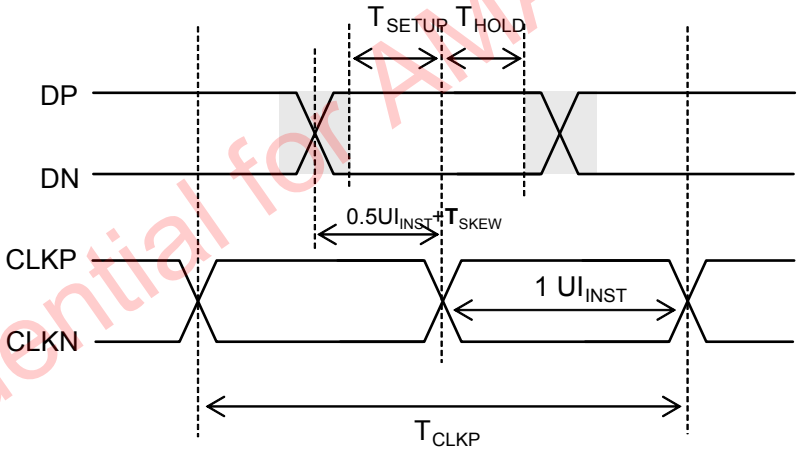


Figure 38 Data to clock timing

Table 31 MIPI clock signal spec

Clock Parameter	Symbol	Min	Nom	Max	Units	Notes
$UI_{instantaneous}$	UI_{INST}	0.8		1.2	UI_{NOM}	
Data to Clock Skew	T_{SKEW}	-0.075		0.075	U_{INOM}	

Electrical Characteristics

Due to having intolerable input pads, the input signal must have HVDD power level for stable operation. If the power of input signal is higher than the recommended level, leakage current may flow via short circuit path in the input pads.

DC Characteristics

Absolute maximum ratings [1]
AVDD supply voltage : -0.3 [V] to 4.0 [V]
HVDD supply voltage : -0.3 [V] to 4.0 [V]
DVDD supply voltage : -0.3 [V] to 1.8 [V]
DC voltage at any input pin : -0.3 [V] to HVDD+0.3 [V]
DC voltage at any output pin : -0.3 [V] to HVDD+0.3 [V]
Storage temperature : -40 [^{o}C] to + 125 [^{o}C]

Table 32 DC characteristics

Symbol	Descriptions	Min	Typ	Max	Unit
AVDD	Analog voltage relative to GND(AGND) level	2.97	3.3	3.63	[V]
HVDD	High VDD(HVDD) voltage relative to GND(DGND) level	1.62	1.8 3.3	3.63	[V]
DVDD	Digital VDD(DVDD) voltage relative to GND(DGND) level	1.152	1.2	1.32	[V]
I_{DDD}	HVDD=3.3 [V] @ Parallel, 30fps	-	8.8	TBD	[mA]
	AVDD=3.3 [V] @ Parallel, 30fps	-	21.3	TBD	
	DVDD=1.2 [V] @ Parallel, 30fps	-	42	TBD	
	HVDD=3.3 [V] @ MIPI, 30fps	-	0	TBD	
	AVDD=3.3 [V] @ MIPI, 30fps	-	21.3	TBD	
	DVDD=1.2 [V] @ MIPI, 30fps	-	53.3	TBD	
I_{DDS}	Standby supply current	-	90	TBD	[uA]
V_{IL1}	Input voltage low level	-	-	HVDD*0.3	[V]
V_{IH1}	Input voltage high level	HVDD*0.7	-	-	[V]
V_{IL2}	Input voltage low level for rClk, rData.	-	-	HVDD*0.3	[V]
V_{IH2}	Input voltage high level for rClk, rData	HVDD*0.7	-	-	[V]
C_{IN}	Input pin capacitance	-	-	10	[pF]
V_{OL1}	Output voltage low	-	-	HVDD*0.2	[V]
V_{OH1}	Output voltage high	HVDD*0.8	-	-	[V]
V_{OL2}	Output voltage low level for rClk, rData.	-	-	HVDD*0.2	[V]
V_{OH2}	Output voltage high level for rData.	HVDD*0.8	-	-	[V]
I_{IN}	Input leakage current	-10	-	10	[uA]
I_{OT}	Output leakage current	-10	-	10	[uA]

[1]Excessive stresses may cause permanent damage to the device.

AC Characteristics

SCL and SDA lines are pulled up to HVDD by 2 Kohm off-chip resistors.

Table 33 2-wire serial interface timing

Symbol	Descriptions	Min	Typ	Max	Unit
f_{SCL}	2-wire serial interface Clock frequency	-	-	400	kHz
T_{ic}	2-wire serial interface Clock period	2.5	-	-	us
T_{icl}	2-wire serial interface Clock low level width	1.66	-	-	us
T_{ich}	2-wire serial interface Clock high level width	0.83	-	-	us
T_{iss}	setup time for start condition	0.83	-	-	us
T_{ihs}	hold time for start condition	0.83	-	-	us
T_{isd}	setup time for input data	266	-	-	ns
T_{ihd}	hold time for input data	0	-	-	ns
T_{isp}	setup time for stop condition	0.83	-	-	us
T_{buf}	bus free time between a stop and a new start condition	1.66	-	-	us
T_{oaa}	delay from SCL falling edge to output data transition	-	-	354	ns
T_r	10% to 90% rising time for SCL/SDA	139	-	336	ns
T_f	90% to 10% falling time for SCL/SDA	4.3	-	5.8	ns
R_p	Off-chip pull-up resistor for SCL/SDA	-	2	-	kohm

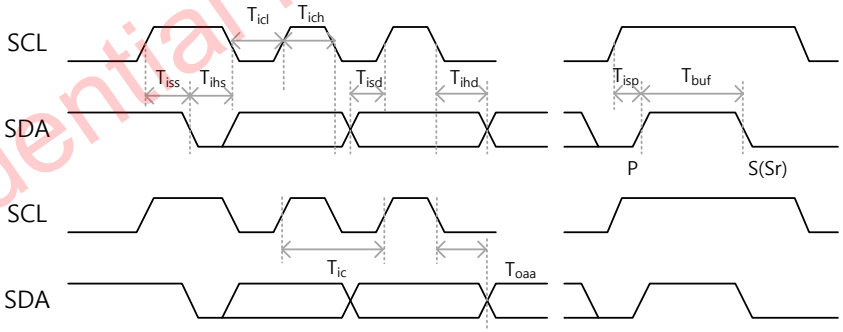


Figure 39 Timing diagram of SCL and SDA

Register Map

Table 34 Register Table - Group A

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
DeviceID_H	A	00	[7:0]	0x21	RO		Device ID High Byte
DeviceID_L	A	01	[7:0]	0x30	RO		Device ID Low Byte
RevNumber	A	02	[7:0]	0x00	RO		Revision number
bank	A	03	[7:0]	0x00	RW		Register group selector
mirror	A	05	[1:0]	0x00	RW	aev	Image Inversion mirror[1] : vertical inversion mirror[0] : horizontal inversion
framewidth_h	A	06	[4:0]	0x0B	RW	aev	Framewidth High Byte (must be larger than window width)
framewidth_l	A	07	[7:0]	0x99	RW	aev	Framewidth Low Byte (must be larger than window width)
fheight_a_h	A	08	[4:0]	0x03	RW	aev	Frameheight High Byte (must be larger than window height)
fheight_a_l	A	09	[7:0]	0xE7	RW	aev	Frameheight Low Byte (must be larger than window height)
i2c_control_1	A	1A	[7:0]	0x50	RW		I2c control register 1
pad_control11	A	21	[7:0]	0x00	RW		Pad control 11
pad_control12	A	22	[7:0]	0x00	RW		Pad control 12
softreset	A	24	[0]	0x00	RW		Soft reset 1'b0 : disable 1'b1 : enable (after succesful reset value reverts to 0)
clkdiv1	A	25	[7:0]	0x1B	RW	aev	Clock divider 1
clkdiv2	A	26	[7:0]	0x10	RW	aev	Clock divider 2
pad_control1	A	27	[7:0]	0x60	RW		Pad control 1
pad_control2	A	28	[7:0]	0x00	RW		Pad control 2
pad_control3	A	29	[7:0]	0x00	RW		Pad control 3
pad_control4	A	2A	[7:0]	0x00	RW		Pad control 4
pad_control5	A	2B	[7:0]	0x00	RW		Pad control 5
pad_control6	A	2C	[7:0]	0x00	RW		Pad control 6
pad_control7	A	2D	[7:0]	0x00	RW		Pad control 7
pad_control8	A	2E	[7:0]	0x00	RW		Pad control 8
pad_control9	A	2F	[7:0]	0x00	RW		Pad control 9
pad_control10	A	30	[7:0]	0x00	RW		Pad control 10
pll_control2	A	48	[7:0]	0x7A	RW		PLL control 2
pll_tg_n_cnt	A	4B	[7:0]	0x2C	RW		PLL multiplication factor for TG
pll_tg_r_cnt	A	4C	[4:0]	0x04	RW		PLL division factor for TG

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
pll_mp_n_cnt	A	4D	[7:0]	0x4D	RW		PLL multiplication factor for MIPI
pll_mp_r_cnt	A	4E	[4:0]	0x04	RW		PLL division factor for MIPI
sync_blankEAV_h	A	58	[3:0]	0x0B	RW		Blanking EAV control High Byte
sync_blankEAV_l	A	59	[7:0]	0x60	RW		Blanking EAV control Low Byte
sync_blankSAV_h	A	5A	[3:0]	0x0A	RW		Blanking SAV control High Byte
sync_blankSAV_l	A	5B	[7:0]	0xB0	RW		Blanking SAV control Low Byte
sync_activeEAV_h	A	5C	[3:0]	0x09	RW		Active EAV control High Byte
sync_activeEAV_l	A	5D	[7:0]	0xD0	RW		Active EAV control Low Byte
sync_activeSAV_h	A	5E	[3:0]	0x08	RW		Active SAV control High Byte
sync_activeSAV_l	A	5F	[7:0]	0x00	RW		Active SAV control Low Byte
sync_CCIR_FF_h	A	60	[3:0]	0x0F	RW		Format header control 0 (FF) High Byte
sync_CCIR_FF_l	A	61	[7:0]	0xFF	RW		Format header control 0 (FF) Low Byte
sync_CCIR_00_h	A	62	[3:0]	0x00	RW		Format header control 1 (00) High Byte
sync_CCIR_00_l	A	63	[7:0]	0x00	RW		Format header control 1 (00) Low Byte
sync_CCIR_80_h	A	64	[3:0]	0x08	RW		Format header control 2 (80) High Byte
sync_CCIR_80_l	A	65	[7:0]	0x00	RW		Format header control 2 (80) Low Byte
sync_CCIR_10_h	A	66	[3:0]	0x01	RW		Format header control 3 (10) High Byte
sync_CCIR_10_l	A	67	[7:0]	0x00	RW		Format header control 3 (10) Low Byte
vsyncstartrow0_h	A	69	[4:0]	0x00	RW	aev	Vertical sync start control for parallel interface High Byte Frame start control for MIPI interface High Byte
vsyncstartrow0_l	A	6A	[7:0]	0x0D	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsyncstoprow0_h	A	6B	[4:0]	0x03	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsyncstoprow0_l	A	6C	[7:0]	0xDD	RW	aev	Vertical sync start control for parallel interface Low Byte Frame start control for MIPI interface Low Byte
vsynccolumn0_h	A	6D	[4:0]	0x00	RW	aev	Internal vsync 0 start point High Byte @ column counter
vsynccolumn0_l	A	6E	[7:0]	0x02	RW	aev	Internal vsync 0 start point Low Byte @ column counter
sync_control_0	A	86	[7:0]	0x80	RW	aev	Sync_control 0
sync_control_1	A	87	[7:0]	0x00	RW	aev	Sync_control 1
data_min_h	A	8E	[3:0]	0x00	RW		Active data control High Byte data_clamp = 1'b1 acive data < data_min

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							output data = data_min
data_min_l	A	8F	[7:0]	0x01	RW		Active data control Low Byte data_clamp = 1'b1 acive data < data_min output data = data_min
data_max_h	A	90	[3:0]	0x00	RW		Active data control High Byte data_clamp = 1'b1 acive data > data_max output data = data_max
data_max_l	A	91	[7:0]	0xFE	RW		Active data control Low Byte data_clamp = 1'b1 acive data > data_max output data = data_max

Table 35 Register Table - Group B

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
genlock_width	B	16	[7:0]	0x10	RW		Genlock puslse width
minline_c	B	17	[1:0]	0x02	RW		Minimum expose
bayer_control_01	B	1B	[7:0]	0x05	RW		Bayer control 1
rcount_genlock_h	B	4B	[4:0]	0x00	RW		Genlock row count High Byte
rcount_genlock_l	B	4C	[7:0]	0x01	RW		Genlock row count Low Byte
ccount_genlock_h	B	4D	[4:0]	0x00	RW		Genlock column count High Byte
ccount_genlock_l	B	4E	[7:0]	0x01	RW		Genlock column count Low Byte
mpu_gen_line_h	B	A5	[4:0]	0x00	RW	m_wr	Frame height resized by mpu_gen_line High Byte
mpu_gen_line_l	B	A6	[7:0]	0x00	RW	m_wr	Frame height resized by mpu_gen_line Low Byte
mpu_gen_col_h	B	A7	[4:0]	0x00	RW	m_wr	Frame width resized by mpu_gen_col High Byte
mpu_gen_col_l	B	A8	[7:0]	0x00	RW	m_wr	Frame width resized by mpu_gen_col Low Byte
mpu_gen_write	B	A9	[0]	0x0	RW		Genlock I2C register update
wr_en_off	B	B3	[0]	0x00	RW		Update exposure related register 1'b0 : update @ wr_en = 1'b1 1'b1 : Immediately update
tp_control_0	B	B4	[7:0]	0x00	RW		Test pattern selection
tp_control_1	B	B5	[7:0]	0x00	RW		R[9:2] color for test pattern
tp_control_2	B	B6	[7:0]	0x00	RW		G1[9:2] color for test pattern
tp_control_3	B	B7	[7:0]	0x00	RW		G2[9:2] color for test pattern
tp_control_4	B	B8	[7:0]	0x00	RW		B[9:2] color for test pattern
tp_control_5	B	B9	[7:0]	0x00	RW		{R[1:0],G1[1:0],G2[1:0],B[1:0]} for test pattern
tp_width_h	B	BA	[2:0]	0x05	RW		Test pattern width for color bar pattern High Byte
tp_width_l	B	BB	[7:0]	0x10	RW		Test pattern width for color bar pattern Low Byte
wr_en	B	BF	[0]	0x00	RW		Update exposure related register 1'b0 : no update 1'b1 : wr_en set
inttime_h	B	C0	[7:0]	0x01	RW	wr_en	Integration time (line) High Byte
inttime_m	B	C1	[7:0]	0x40	RW	wr_en	Integration time (line) Low Byte
inttime_l	B	C2	[7:0]	0x00	RW	wr_en	Iinteration time (column)
globalgain	B	C3	[7:0]	0x00	RW	wr_en	Analog gain
digitalgain_l	B	C4	[7:0]	0x10	RW	wr_en	Digital gain of long data
digitalgain_s	B	C5	[7:0]	0x10	RW	wr_en	Digital gain of short data

Table 36 Register Table - Group F

Register name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
mipi_control_0	F	04	[7:0]	0x30	RW	aev	MIPI control 0
mipi_control_1	F	05	[7:0]	0xAB	RW		MIPI control 1
mipi_control_2	F	06	[7:0]	0xAA	RW		MIPI control 2
mipi_control_3	F	07	[7:0]	0xAA	RW		MIPI control 3
mipi_control_4	F	08	[7:0]	0x01	RW		MIPI control 4
mipi_control_5	F	09	[7:0]	0x1B	RW		MIPI control 5
mipi_control_6	F	0A	[7:0]	0x00	RW		MIPI control 6
mipi_control_7	F	0B	[7:0]	0x00	RW		MIPI control 7
mipi_test_d0	F	2D	[7:0]	0xAA	RW		MIPI test data 0 for HS state
mipi_test_d1	F	2E	[7:0]	0xFF	RW		MIPI test data 1 for HS state
mipi_test_d2	F	2F	[7:0]	0x00	RW		MIPI test data 2 for HS state
mipi_test_d3	F	30	[7:0]	0x55	RW		MIPI test data 3 for HS state
mipi_pkt_size0_h	F	36	[7:0]	0x09	RW		MIPI word counter size 0 control for image data High Byte
mipi_pkt_size0_l	F	37	[7:0]	0x6F	RW		MIPI word counter size 0 control for image data Low Byte
mipi_data_id0	F	42	[7:0]	0x2B	RW		MIPI data 0 identifier

Table 37 Register Table - Control Registers

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
pad_control1	A	27	[7]	1'b0	RW		Fix to 1'b0
	A	27	[6]	1'b1	RW		reserved
	A	27	[5:4]	2'b10	RW		Output data pad stdby level selector 2'b00 : low 2'b01 : high 2'b1x : hiz
	A	27	[3]	1'b0	RW		Clock pad kill enable 1'b0 : disable (not kill) 1'b1 : enable (kill)
	A	27	[2:0]	3'b000	RW		reserved
pad_control2	A	28	[7:6]	2'b00	RW		Data pad drivability control
	A	28	[5:4]	2'b00	RW		PCLK pad drivability control
	A	28	[3:0]	4'b0000	RW		PCLK timing delay delay = dly_digi_PCLK*0.4 ns
pad_control3	A	29	[7]	1'b0	RW		Vsync pad enable 1'b0 : disable 1'b1 : enable
	A	29	[6:5]	2'b00	RW		Hsync Pad drivability control
	A	29	[4]	1'b0	RW		Hsync pad enable 1'b0 : disable 1'b1 : enable
	A	29	[3]	1'b0	RW		PCLK pad enable 1'b0 : disable 1'b1 : enable
	A	29	[2]	1'b0	RW		Change PCLK phase
	A	29	[1]	1'b0	RW		Data pad swap option 1'b0 : [MSB:LSB] 1'b1 : [LSB:MSB]
	A	29	[0]	1'b0	RW		GENLOCK pad enable 1'b0 : disable 1'b1 : enable
pad_control4	A	2A	[7]	1'b0	RW		D11 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[6]	1'b0	RW		D10 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[5]	1'b0	RW		D9 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[4]	1'b0	RW		D8 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[3]	1'b0	RW		D7 pad control 1'b0 : disable

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							1'b1 : enable
	A	2A	[2]	1'b0	RW		D6 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[1]	1'b0	RW		D5 pad control 1'b0 : disable 1'b1 : enable
	A	2A	[0]	1'b0	RW		D4 pad control 1'b0 : disable 1'b1 : enable
pad_control5	A	2B	[7]	1'b0	RW		D3 pad control 1'b0 : disable 1'b1 : enable
	A	2B	[6]	1'b0	RW		D2 pad control 1'b0 : disable 1'b1 : enable
	A	2B	[5]	1'b0	RW		D1 pad control 1'b0 : disable 1'b1 : enable
	A	2B	[4]	1'b0	RW		D0 pad control 1'b0 : disable 1'b1 : enable
	A	2B	[3]	1'b0	RW		reserved
	A	2B	[2]	1'b0	RW		reserved
	A	2B	[1]	1'b0	RW		reserved
	A	2B	[0]	1'b0	RW		reserved
pad_control6	A	2C	[7]	1'b0	RW		reserved
	A	2C	[6:4]	3'b000	RW		HSYNC timing delay delay = dly_hsync_ctrl*0.8 ns
	A	2C	[3]	1'b0	RW		reserved
	A	2C	[2:0]	3'b000	RW		D11 timing delay delay = dly_hsync_ctrl*0.8 ns
pad_control7	A	2D	[7]	1'b0	RW		reserved
	A	2D	[6:4]	3'b000	RW		D10 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	2D	[3]	1'b0	RW		reserved
	A	2D	[2:0]	3'b000	RW		D9 timing delay delay = dly_hsync_ctrl*0.8 ns
pad_control8	A	2E	[7]	1'b0	RW		reserved
	A	2E	[6:4]	3'b000	RW		D8 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	2E	[3]	1'b0	RW		reserved
	A	2E	[2:0]	3'b000	RW		D7 timing delay delay = dly_hsync_ctrl*0.8 ns

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
pad_control9	A	2F	[7]	1'b0	RW		reserved
	A	2F	[6:4]	3'b000	RW		D6 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	2F	[3]	1'b0	RW		reserved
	A	2F	[2:0]	3'b000	RW		D5 timing delay delay = dly_hsync_ctrl*0.8 ns
pad_control10	A	30	[7]	1'b0	RW		reserved
	A	30	[6:4]	3'b000	RW		D4 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	30	[3]	1'b0	RW		reserved
	A	30	[2:0]	3'b000	RW		D3 timing delay delay = dly_hsync_ctrl*0.8 ns
pad_control11	A	21	[7]	1'b0	RW		reserved
	A	21	[6:4]	3'b000	RW		D2 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	21	[3]	1'b0	RW		reserved
	A	21	[2:0]	3'b000	RW		D1 timing delay delay = dly_hsync_ctrl*0.8 ns
pad_control12	A	22	[7]	1'b0	RW		reserved
	A	22	[6:4]	3'b000	RW		D0 timing delay delay = dly_hsync_ctrl*0.8 ns
	A	22	[3]	1'b0	RW		reserved
	A	22	[2:0]	3'b000	RW		reserved
i2c_control_1	A	1A	[7:4]	4'b0101	RW		Control I2C register with autov, aev update type LSB 2-bit updatecontrol[1:0] controls aev update and MSB 2-bit updatecontrol[3:2] controls autov update 2'b00 : no update 2'b01 : aev update 2'b1x : immediate aev update
	A	1A	[3]	1'b0	RW		reserved
	A	1A	[2]	1'b0	RW		reserved
	A	1A	[1]	1'b0	RW		reserved
	A	1A	[0]	1'b0	RW		reserved
clkdiv1	A	25	[7:6]	2'b00	RW	aev	ADC clock divider isp_clk =vco1/(2^adcclk_div)
	A	25	[5:4]	2'b01	RW	aev	ISP clock divider isp_clk =vco1/(2^ispclk_div)
	A	25	[3:2]	2'b10	RW	aev	PLL2 input clock divider isp_clk =isp_clk/(2^pll2_in_div)
	A	25	[1]	1'b1	RW	aev	PLL2 input clock selector 1'b0 : pll2_in_div output clock

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							1'b1 : main OSC clock
	A	25	[0]	1'b1	RW	aev	pclk clock divider pclk = isp_clk/(2^pclk_div)
clkdiv2	A	26	[7:6]	2'b00	RW	aev	MIPI clock divider mipi_clk =vco2/(2^mipiclk_div)
	A	26	[5:3]	3'b010	RW	aev	MIPI byte clock divider ddclk =vco2/(2^ddclk_div)
	A	26	[2]	1'b0	RW	aev	reserved
	A	26	[1]	1'b0	RW	aev	reserved
	A	26	[0]	1'b0	RW	aev	reserved
pll_control2	A	48	[7]	1'b0	RW		reseved
	A	48	[6]	1'b1	RW		Fix to 'b1
	A	48	[5]	1'b1	RW		PLL1 power down mode 1'b0 : pll1 power on 1'b1 : pll1 power down
	A	48	[4]	1'b1	RW		PLL bypass 1'b0 : use pll mode 1'b1 : pll bypass mode
	A	48	[3]	1'b1	RW		PLL2 power down mode 1'b0 : pll2 power on 1'b1 : pll2 power down
	A	48	[2]	1'b0	RW		reserved
	A	48	[1:0]	2'b10	RW		TG VCO oscillation band selection 2'b00 : OFF, 2'b01 : LB, 2'b10 : MB, 2'b11 : HB
bayer_control_01	B	1B	[7]	1'b0	RW		reserved
	B	1B	[6]	1'b0	RW		Variable frame enable 1'b0 : disable 1'b1 : enable
	B	1B	[5]	1'b0	RW		GENLOCK enable 1'b0 : disable 1'b1 : enable
	B	1B	[4]	1'b0	RW		GENLOCK master 1'b0 : slave 1'b1 : master
	B	1B	[3:2]	2'b01	RW		reserved
	B	1B	[1:0]	2'b01	RW		reserved
sync_control_0	A	86	[7]	1'b1	RW	aev	reserved
	A	86	[6:5]	2'b00	RW	aev	Vsync, hsync drop control 2'b00 : No drop 2'b01 : vsync drop 2'b10 : hsync drop 2'b11 : hsync and vsync drop
	A	86	[4]	1'b0	RW	aev	reserved

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
	A	86	[3]	1'b0	RW	aev	reserved
	A	86	[2]	1'b0	RW	aev	reserved
sync_control_1	A	87	[7]	1'b0	RW	aev	reserved
	A	87	[6]	1'b0	RW	aev	Vsync polarity change 1'b0 : disable 1'b1 : enable
	A	87	[5]	1'b0	RW	aev	Hsync output all lines enable(black and active) 1'b0 : No hsync during vertical blank 1'b1 : hsync during vertical blank
	A	87	[4]	1'b0	RW	aev	Hsync polarity change 1'b0 : disable 1'b1 : enable
	A	87	[3]	1'b0	RW	aev	reserved
	A	87	[2]	1'b0	RW	aev	reserved
	A	87	[1]	1'b0	RW	aev	Effective data clamping enable 1'b0 : disable 1'b1 : enable
	A	87	[0]	1'b0	RW	aev	reserved
mipi_control_0	F	04	[7]	1'b0	RW	aev	reserved
	F	04	[6]	1'b0	RW	aev	MIPI enable 1'b0 : disable 1'b1 : enable
	F	04	[5]	1'b1	RW	aev	MIPI DPHY global power down 1'b0 : disable 1'b1 : enable
	F	04	[4]	1'b1	RW	aev	MIPI clock lane hs mode 1'b0 : LP & HS mode 1b'1 : only HS mode
	F	04	[3]	1'b0	RW	aev	MIPI ULP data pad 1'b0 : ultra low power mode disable 1'b1 : ultra low power mode enable
	F	04	[2]	1'b0	RW	aev	MIPI ULP clock pad 1'b0 : ultra low power mode disable 1'b1 : ultra low power mode enable
	F	04	[1]	1'b0	RW	aev	reserved
	F	04	[0]	1'b0	RW	aev	Fix to 1'b0
mipi_control_1	F	05	[7:4]	4'b1010	RW		MIPI CKP/CKN pad state control 4'b0000 : Normal operation mode 4'b0001 : CKP/CKN = LP-00 state 4'b0010 : CKP/CKN = LP-01 state 4'b0011 : CKP/CKN = LP-10 state 4'b0100 : CKP/CKN = LP-11 state 4'b0101 : CKP/CKN = HS-0 state 4'b0110 : CKP/CKN = HS-1 state 4'b0111 : CKP/CKN = Hi-z state

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							4'b1000 : CKP/CKN = ULP state 4'b1010 : CKP/CKN = power down
	F	05	[3:2]	2'b10	RW		mipi CP/CN control
	F	05	[1]	1'b1	RW		reserved
	F	05	[0]	1'b1	RW		reserved
mipi_control_2	F	06	[7:4]	4'b1010	RW		MIPI DP0/DN0 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP0/DN0 = LP-00 state 4'b0010 : DP0/DN0 = LP-01 state 4'b0011 : DP0/DN0 = LP-10 state 4'b0100 : DP0/DN0 = LP-11 state 4'b0101 : DP0/DN0 = HS-0 state 4'b0110 : DP0/DN0 = HS-1 state 4'b0111 : DP0/DN0 = Hi-z state 4'b1000 : DP0/DN0 = ULP state 4'b1001 : DP0/DN0 = Serializer Test mode 4'b1010 : DP0/DN0 = power down
	F	06	[3:0]	4'b1010	RW		MIPI DP1/DN1 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP1/DN1 = LP-00 state 4'b0010 : DP1/DN1 = LP-01 state 4'b0011 : DP1/DN1 = LP-10 state 4'b0100 : DP1/DN1 = LP-11 state 4'b0101 : DP1/DN1 = HS-0 state 4'b0110 : DP1/DN1 = HS-1 state 4'b0111 : DP1/DN1 = Hi-z state 4'b1000 : DP1/DN1 = ULP state 4'b1001 : DP1/DN1 = Serializer Test mode 4'b1010 : DP1/DN1 = power down
mipi_control_3	F	07	[7:4]	4'b1010	RW		MIPI DP2/DN2 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP2/DN2 = LP-00 state 4'b0010 : DP2/DN2 = LP-01 state 4'b0011 : DP2/DN2 = LP-10 state 4'b0100 : DP2/DN2 = LP-11 state 4'b0101 : DP2/DN2 = HS-0 state 4'b0110 : DP2/DN2 = HS-1 state 4'b0111 : DP2/DN2 = Hi-z state 4'b1000 : DP2/DN2 = ULP state 4'b1001 : DP2/DN2 = Serializer Test mode 4'b1010 : DP2/DN2 = power down
	F	07	[3:0]	4'b1010	RW		MIPI DP3/DN3 pad state control 4'b0000 : Normal operation mode 4'b0001 : DP3/DN3 = LP-00 state 4'b0010 : DP3/DN3 = LP-01 state 4'b0011 : DP3/DN3 = LP-10 state 4'b0100 : DP3/DN3 = LP-11 state 4'b0101 : DP3/DN3 = HS-0 state

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
							4'b0110 : DP3/DN3 = HS-1 state 4'b0111 : DP3/DN3 = Hi-z state 4'b1000 : DP3/DN3 = ULP state 4'b1001 : DP3/DN3 = Serializer Test mode 4'b1010 : DP3/DN3 = power down
mipi_control_4	F	08	[7:6]	2'b00	RW		Number of virtural channel 2'b00 : 1 2'b01 : 2 2'b1x : not use
	F	08	[5:4]	2'b00	RW		Data seletor of the mipi_format block 2'b01 : mipi data 12 bit
	F	08	[3:2]	2'b00	RW		MIPI lane selector 2'b01 : 2 lane mode 2'b10 : 4 lane mode
	F	08	[1:0]	2'b01	RW		phy sync control register
mipi_control_5	F	09	[7:6]	2'b00	RW		Data0 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
	F	09	[5:4]	2'b01	RW		Data1 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
	F	09	[3:2]	2'b10	RW		Data2 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
	F	09	[1:0]	2'b11	RW		Data3 lane data swap 2'b00 : d0 lane data 2'b01 : d1 lane data 2'b10 : d2 lane data 2'b11 : d3 lane data
mipi_control_6	F	0A	[7]	1'b0	RW		Fix to 1'b0
	F	0A	[6]	1'b0	RW		Fix to 1'b0
	F	0A	[5]	1'b0	RW		Fix to 1'b0
	F	0A	[4]	1'b0	RW		Fix to 1'b0
	F	0A	[3]	1'b0	RW		Fix to 1'b0
	F	0A	[2]	1'b0	RW		Fix to 1'b0
	F	0A	[1:0]	2'b00	RW		reserved
mipi_control_7	F	0B	[7]	1'b0	RW		Data0 N/P swap
	F	0B	[6]	1'b0	RW		Data1 N/P swap
	F	0B	[5]	1'b0	RW		Data2 N/P swap

Register Name	Address		Bits	Init. Val.	Type	Up date	Description
	Bank	Hex					
	F	0B	[4]	1'b0	RW		Data3 N/P swap
	F	0B	[3]	1'b0	RW		Mipi clock N/P swap
	F	0B	[2]	1'b0	RW		reserved
	F	0B	[1:0]	2'b00	RW		reserved

Revision History

Version	Date [D/M/Y]	Notes	Writer
0.0	07/06/2019	(Preliminary)	Kibbeum Kim
0.1	27/09/2019	• Update Key parameter table. - P.5 – Dark signal – Sensitivity – HVDD – Dynamic range – SNR – Package Type • Update Absolute maximum rating of DC Characteristics. - P.44 – AVDD, HVDD, DVDD	Kibbeum Kim
0.2	04/02/2020	• Update Key parameter table. - P.5~6 – Power consumption – Sensitivity – Package Type • Update PLL VCO minimum, maximum value. - P.18 • Update DC Characteristics. - P.44 – IDDD – IDDS • Update AC Characteristics.- P.45 – Rp	Kibbeum Kim
0.3	20/02/2020	• Update Key parameter table. - P.5~6 – Power consumption	Kibbeum Kim
0.4	31/03/2020	• Update Register Table - P.49	Kibbeum Kim
0.5	22/04/2020	• Update Key parameter table. - P.5 – Maximum frame rate	Kibbeum Kim
0.6	11/05/2020	• Update Key parameter table. - P.6 – HVDD • Update DC Characteristics. - P.44 – HVDD – DVDD	Jaedong Park
0.7	08/06/2020	• Update Key parameter table. - P.5 – Operating Temp. – Max. frame rate	Jaedong Park
0.8	13/07/2020	• Update Revision history. – deleted Ver 0.4 history • Update DC characteristics. - P.44 – Table format • Update AC characteristics. - P.45 – Table format	Jaedong Park