S5K5BAFX

1/5” 2Mp CMOS Image Sensor SoC with an Embedded Image Processor

Technical Data Sheet

(Preliminary – EVT2, R01)

REVISION HISTORY

Revision	Date	Amendment
R00	19-Aug-08	Initial draft
R01	13-Oct-08	

Contents

Revision History 2
Features 4
General Description 5
Logical Symbol Diagram 6
Pad Configuration 7
Pad Description 8
Pixel Array Information 11
Video Output Interface Description 12
Control Interface Description 21
Functional Description 26
System State Diagram 36
Power-Up/Down Sequence 37
Standby Sequence 38
Electrical Characteristics 39
Imaging Characteristics 40
Register Description 41

FEATURES

Image Sensor

- Optical format : 1/5 inch
- Unit pixel size : 1.75um
- Effective resolution : 1600 (H) x 1200 (V)
- Active resolurion : 1616 (H) x 1216 (V)
- Color filter : RGB Bayer pattern
- Shutter type : Electronic rolling shutter
- Max. capture frame rate : 15fps @full resolution
- Max. video frame rate : 30fps @VGA
- Max. pixel clock Frequency : 70MHz
- Max. pixel rate : 35Mp/s
- ADC accuracy : 10-bit
- Progressive scan readout
- Window panning & cropping
- Vertical flip and horizontal mirror mode
- Continuous and single frame capture mode
- Frame rate control
- 2x2 analog averaged sub-sampling
- LED and flash strobe mode
- Parallel output format: ITU-R. 656/601 YUV422, RGB888/RGB565, RAW10
- Serial output format: MIPI CSI2 (single lane) YUV422, RGB888/RGB565, RAW10

Image Processor

- Color recovery and correction
- False color suppression
- Lens shading correction
- Noise removal
- Edge enhancement
- SXGA or any size smaller than SXGA down scaling
- Programmable gamma correction
- Auto defect correction
- Auto dark level compensation
- Auto flicker correction (50/60Hz)
- Auto exposure (AE)
- Auto white balance (AWB)
- Built-in test image generation

Device

- Host control interface: I^2C bus
- Internal PLL (6MHz to 27MHz input frequency)
- Stand-by mode for power saving
- Operating temperature: -20°C to +60°C
- Supply voltage: 2.8V for analog, 1.5V for digital core (with internal regulator off), 1.8V - 2.8V for I/O
- 1.8V to 1.5V internal regulator

GENERAL DESCRIPTION

The S5K5BAFX is a highly integrated 2Mp camera chip which includes CMOS image sensor, image processor and both 8-bit ITU-R 656/601 parallel interface and MIPI CSI2 compliant serial interface. It is fabricated by SAMSUNG 0.13μm CMOS image sensor process developed for imaging application to realize high-efficiency and low-power photo sensor.
The CMOS image sensor consists of 1616x1216 Active Pixel Sensor (APS) array which has 1/5 inch optical format, on-chip 10-bit ADC array to digitize the analog pixel output, and on-chip Correlated Double Sampling (CDS) to reduce Fixed Pattern Noise (FPN) drastically.
The image processor performs sophisticated image processing functions including color recovery and correction, false color suppression, lens shading correction, noise removal, edge enhancement, programmable gamma correction, image down scaling, auto defect correction, auto dark level compensation, auto flicker correction (50/60Hz), auto exposure (AE), auto white balance (AWB). The auto functions are performed by F/W on an embedded RISC processor. The host controller is able to access and control this device via general IIC bus.

LOGICAL SYMBOL DIAGRAM

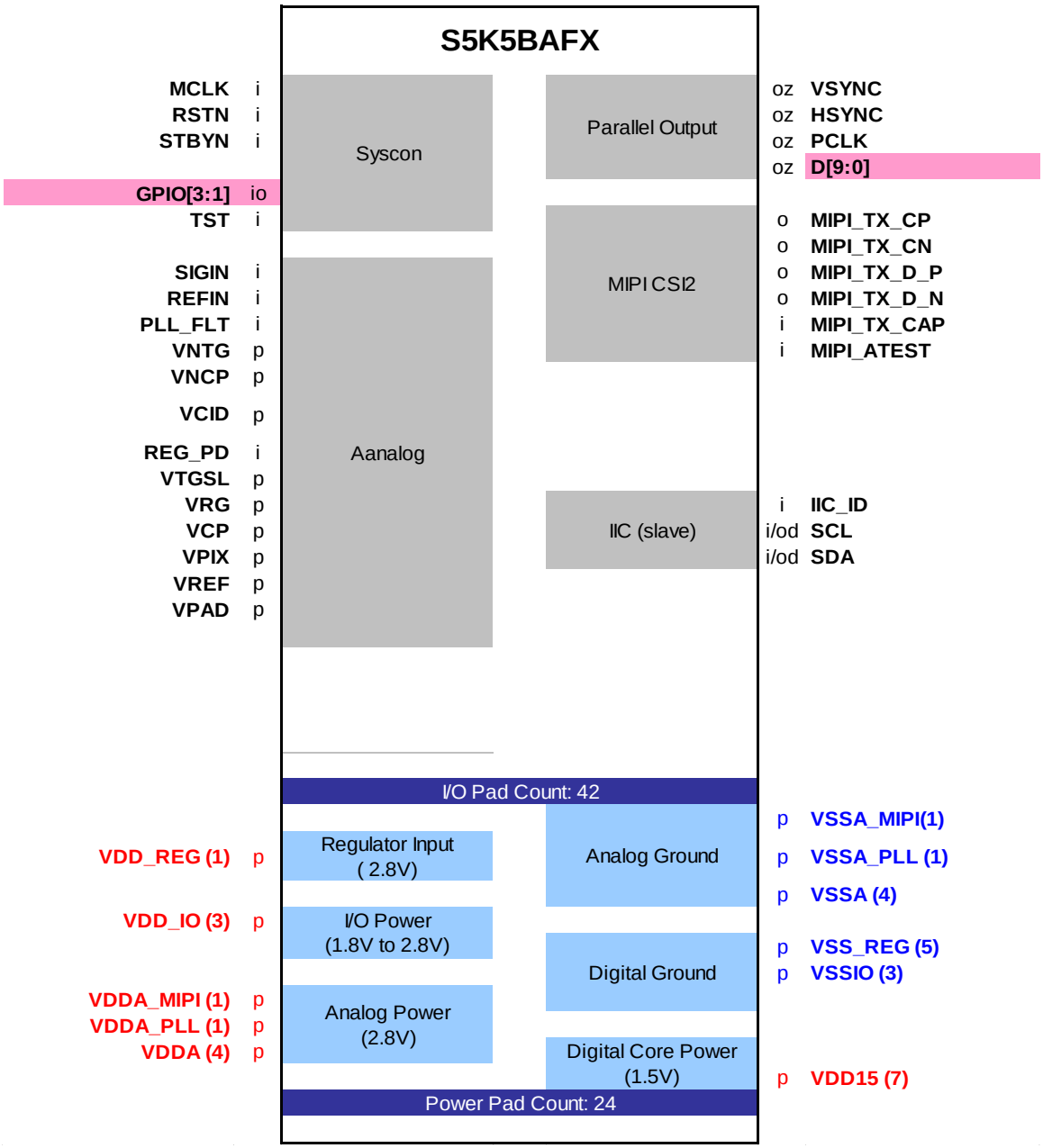


Figure 1: Logical Symbol Diagram

PAD CONFIGURATION

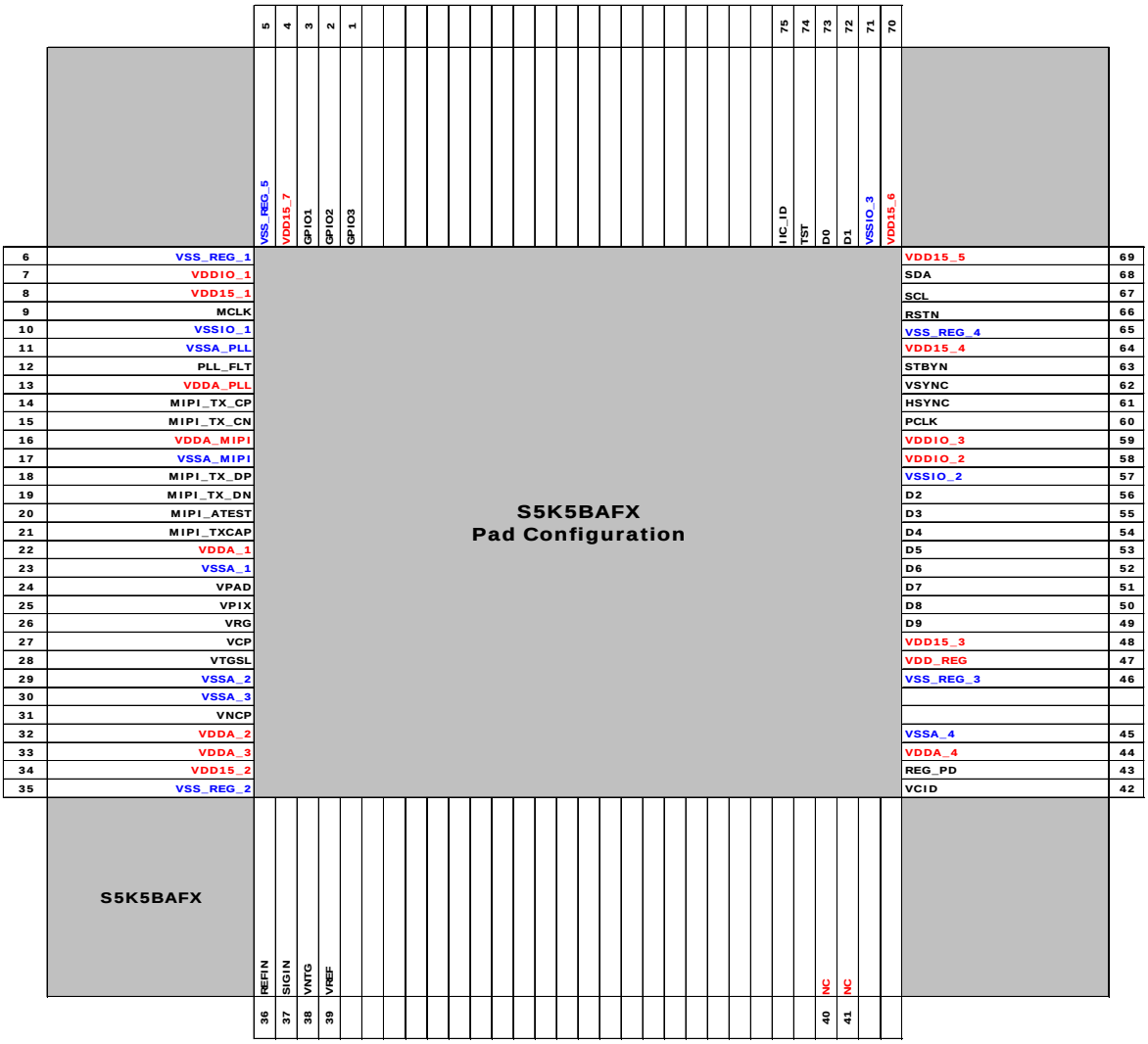


Figure 2: Pad Configuration

PAD DESCRIPTION

Table 1: Pad Description

Pad No	Pad Name	I/O	Description
9	**MCLK**	i	Master clock. 6M to 27MHz. 27MHz is default. If lower frequencies are used, the PLL register settings should be changed.
66	**RSTN**	i	Master reset (Active low)
63	**STBYN**	i	Hardware standby mode (Active low). Set to '1' if not used. All parallel outputs go to Hi-Z state during STBYN is asserted.
74	**TST**	i	Test mode. Set to '0' in normal mode
43	**REG_PD**	a	Internal regulator power-down 0:operation, 1:power-down
24	**VPAD**	a	Analog test. Open in normal mode
36	**REFIN**	a	Analog test. Open in normal mode
39	**VREF**	a	Analog pad. 0.1uF external capacitor between pin and ground.
28	**VTGSL**	a	Analog pad. 0.1uF external capacitor between pin and ground.
38	**VNTG**	a	Analog pad. 0.1uF external capacitor between pin and ground.
42	**VCID**	a	Analog test. Connect to 2.8V
12	**PLL_FLT**	a	PLL test. Open in normal
25	**VPIX**	a	Analog test.
26	**VRG**	a	Analog test.
27	**VCP**	a	Analog test.
31	**VNCP**	a	Analog test.
37	**SIGIN**	a	Analog test.
75	**IIC_ID**	i	IIC slave address selection IIC_ID= 0: 0111_100b, 1: 0101_101b
67	**SCL**	iod	IIC slave clock for host control
68	**SDA**	iod	IIC slave data for host control
3	**GPIO_1**	ioz	General purpose I/Os [NOTE] GPIO_1: flash strobe output GPIO_2: flash strobe input
2	**GPIO_2**	ioz	
1	**GPIO_3**	ioz	
62	**VSYNC**	oz	Vertical sync output for parallel interface
61	**HSYNC**	oz	Horizontal sync output for parallel interface
60	**PCLK**	oz	Pixel clock output for parallel interface
73	**D0**	oz	Pixel data output for parallel interface.

72	**D1**	oz	D9: MSB, D0: LSB D[9:0] for 10-bit data D[9:2] for 8-bit data
56	**D2**	oz	
55	**D3**	oz	
54	**D4**	oz	
53	**D5**	oz	
52	**D6**	oz	
51	**D7**	oz	
50	**D8**	oz	
49	**D9**	oz	
14	**MIPI_TX_CP**	o	CSI-2 Tx clock positive. Open if not used.
15	**MIPI_TX_CN**	o	CSI-2 Tx clock negative. Open if not used.
18	**MIPI_TX_DP**	o	CSI-2 Tx data positive. Open if not used.
19	**MIPI_TX_DN**	o	CSI-2 Tx data negative. Open if not used.
20	**MIPI_TX_ATEST**	a	Analog test. Open in normal mode or if not used.
21	**MIPI_TX_CAP**	a	CSI-2 Tx capacitor. 0.1uF external capacitor between pin and ground. Open if not used.
8 34 48 65 69 70 5	**VDD15_1** **VDD15_2** **VDD15_3** **VDD15_4** **VDD15_5** **VDD15_6** **VDD15_7**	p	Digital Core Power **1.5V** (1.4V to 1.6V) [NOTE] a) Regulator on (REG_PD=0): **0.4uF** capacitor between VDD15 and ground b) Regulator off (REG_PD=1): **1.5V** with **0.4uF** power capacitor
47	**VDD_REG**	p	Regulator input power **1.8V** (1.7V to 1.9) [NOTE] a) Regulator on (REG_PD=0): **1.8V** b) Regulator off (REG_PD=1): **1.5V**
7 58 59	**VDDIO_1** **VDDIO_2** **VDDIO_3**	p	I/O power **1.8V** (1.65V to 1.95V) or **2.8V** (2.5V to 3.1V) with **0.1uF** power capacitor

4 6 35 46 64	**VSS_REG_5** **VSS_REG_1** **VSS_REG_2** **VSS_REG_3** **VSS_REG_4**	p	Digital ground
10 57 71	**VSSIO_1** **VSSIO_2** **VSSIO_3**		
16	**VDDA_MIPI**	p	Analog power **2.8V** (2.6V to 3.0V) with **0.1uF** power capacitor
13	**VDDA_PLL**		
22 32 33 44	**VDDA_1** **VDDA_2** **VDDA_3** **VDDA_4**		
17	**VSSA_MIPI**	p	Analog ground
11	**VSSA_PLL**		
23 29 30 45	**VSSA_1** **VSSA_2** **VSSA_3** **VSSA_4**		

PIXEL ARRAY INFORMATION

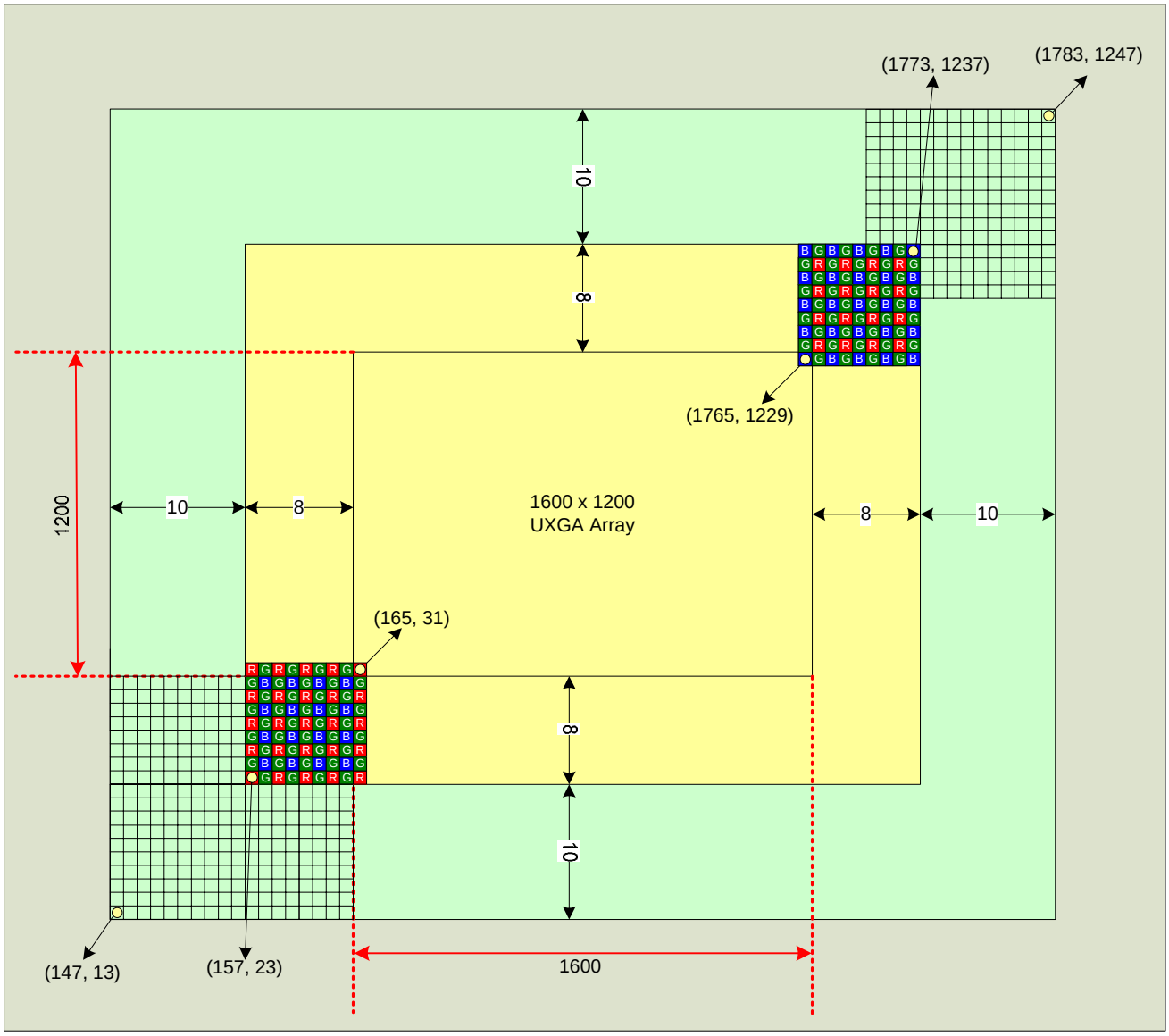
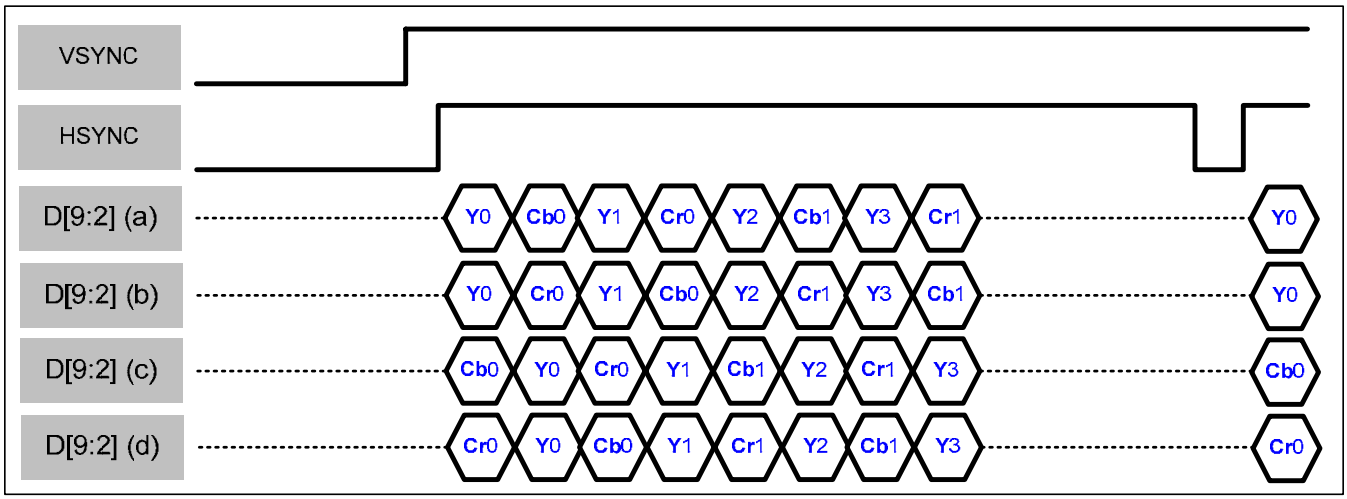


Figure 3: Pixel Arrary Information

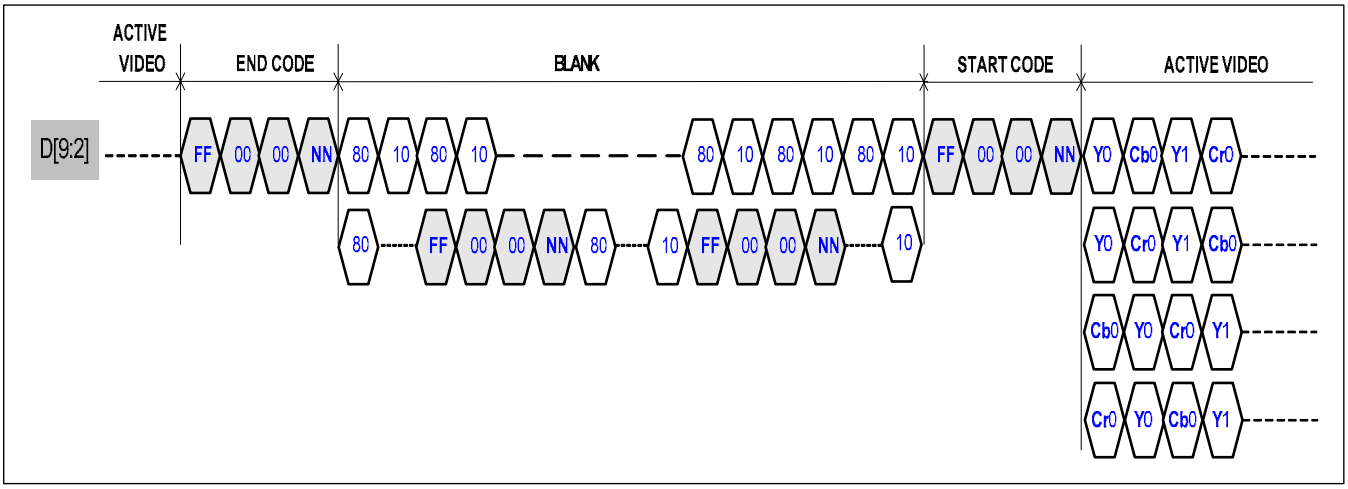
VIDEO OUTPUT INTERFACE DESCRIPTION

Parallel Output Interface



[NOTE] The data output sequence, (a) to (d) can be selected by register setting.

Figure 4: ITU-R.601 YCbCr Data Output Timing



[NOTE]
(1) The video data is in compliance with Recommendation 656.
(2) The data words 0 and 255 (00 and FF in hex notation) are reserved for data identification purposes and consequently only 254 of the possible 256 words may be used to express a signal value.
(3) Each timing reference code consists of a four word sequence in the following format: FF 00 00 NN.
(4) The fourth word(NN) contains information, the state of field blanking, and the state of line blanking
(5) NN consist of 1(MSB, fixed), F, V, H, P3, P2, P1, P0(LSB) bits
(F = 0 during field 1, 1 during field 2, V = 0 elsewhere, 1 during field blanking,
H = 0 in SAV(Start of Active Video), 1 in EAV(End of Active Video), P3,P2,P1,P0 : protection bits)

Figure 5: ITU-R.656 YCbCr Data Output Timing

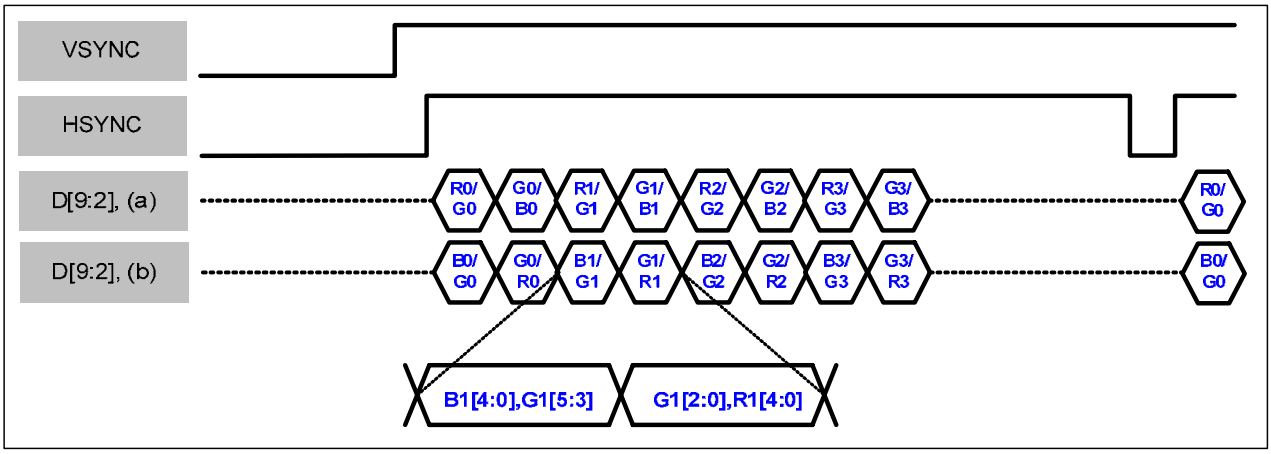


Figure 6: 565RGB Data Output Timing

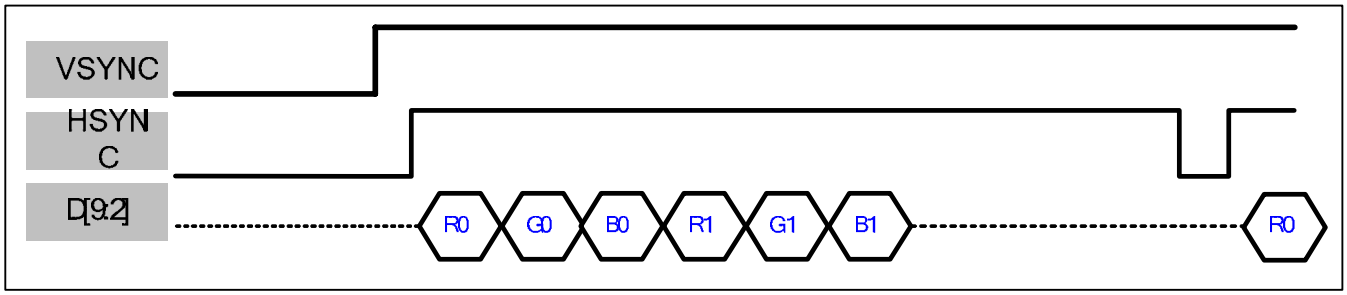
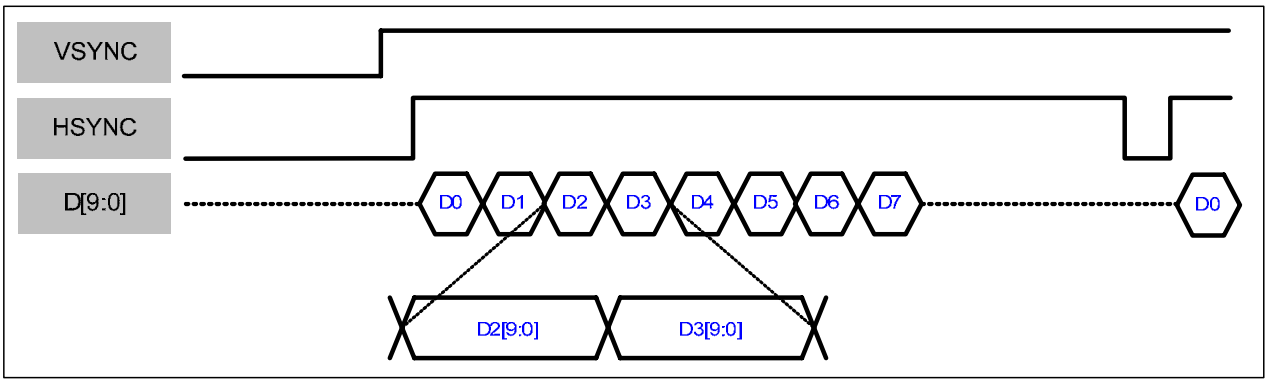


Figure 7: 888RGB Data Output Timing



[NOTE] 10-bit parallel data pads should be bonded for RAW10 interface.

Figure 8: CIS Raw Data Output Timing – RAW10

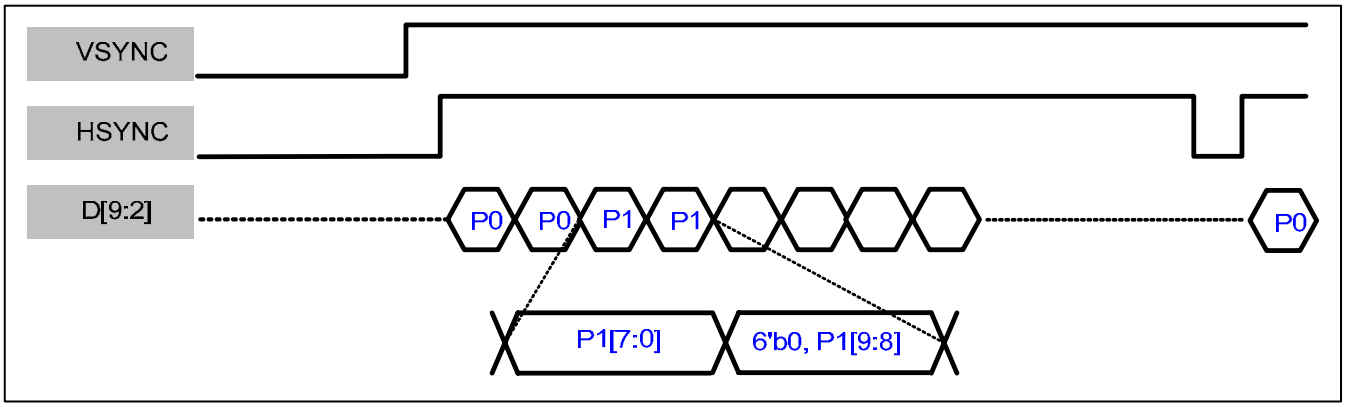
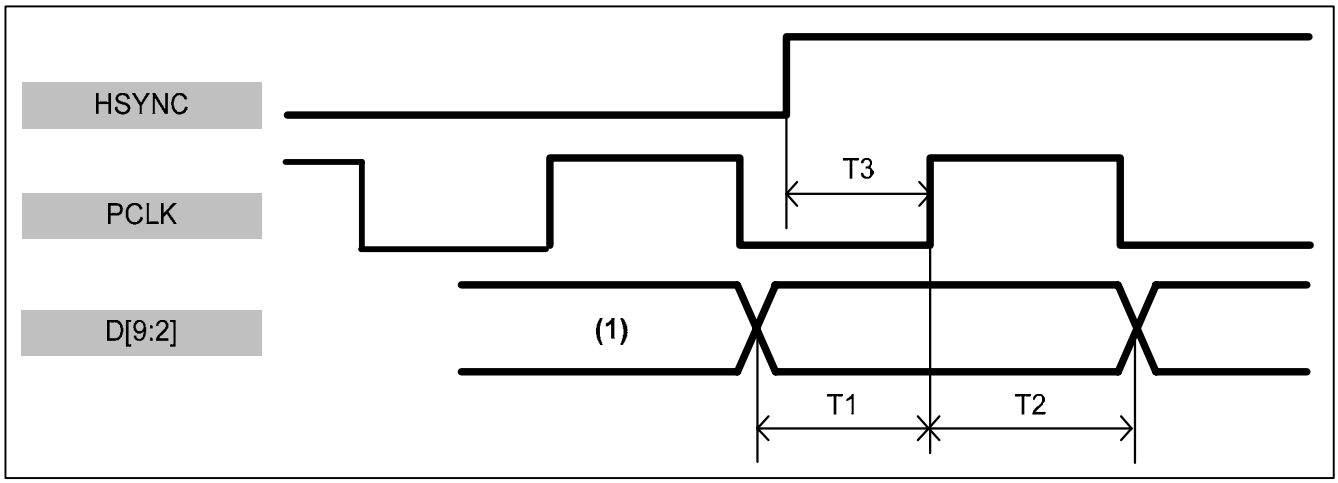


Figure 9: CIS Raw Data Output Timing – RAW10(8+2)



[NOTE] (1): blank & start code, otherwise ‘0’ for ITU-R.656 output format:

SYMBOL	PARAMETER	MIN	MAX	UNIT
T1	Data Setup Time to PCLK	4	-	ns
T2	Data Hold Time to PCLK	4	-	ns
T3	HSYNC↑ to PCLK↑ delay	4	-	ns

Figure 10: Output Data and Pixel Clock Timing

Serial Output Interface (MIPI CSI-2)

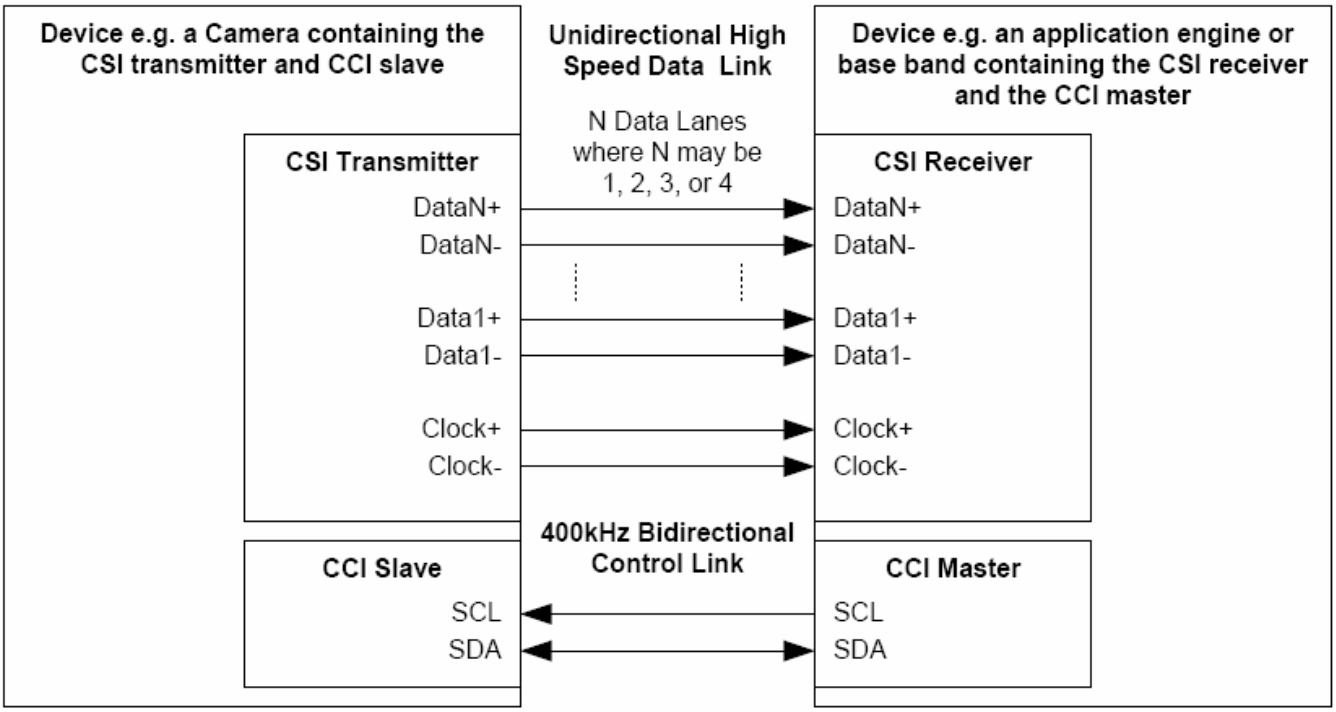


Figure 11: CSI-2 and CCI Transmitter and Receiver Interface

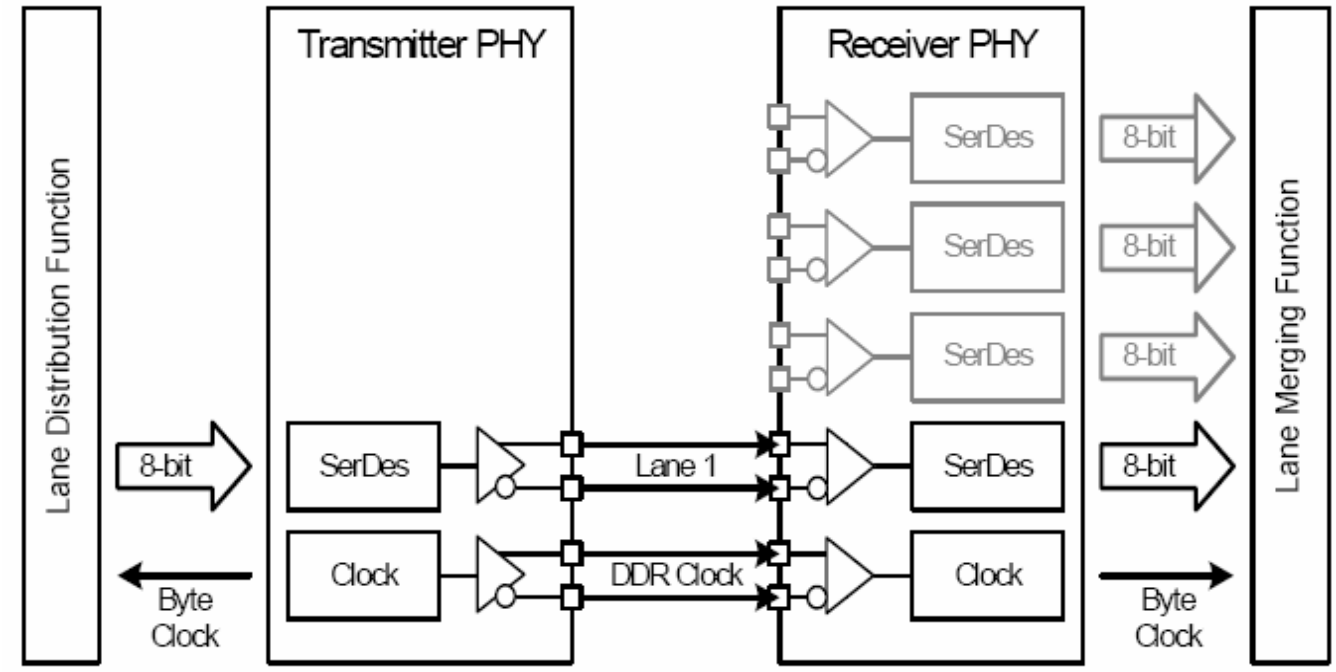


Figure 12: One Lane Transmitter and Four Lane Receiver Example

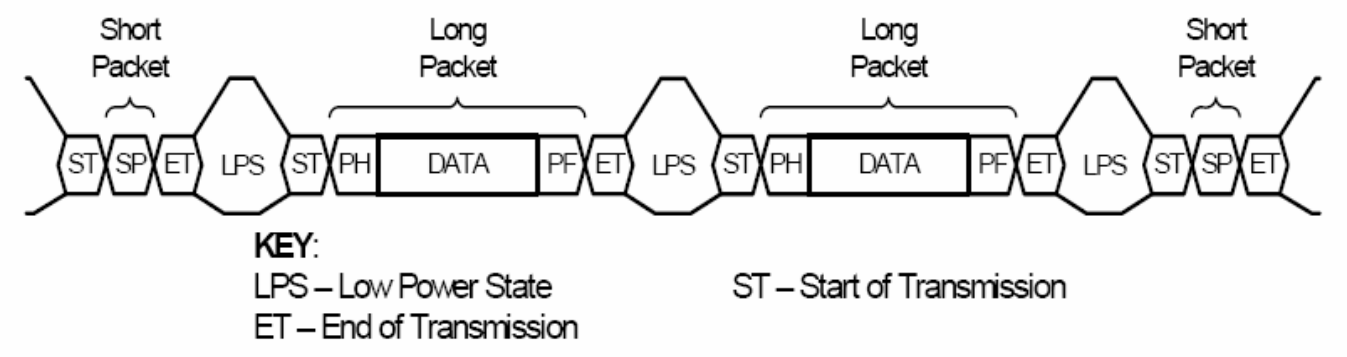


Figure 13: Low Level Protocol Packet Overview

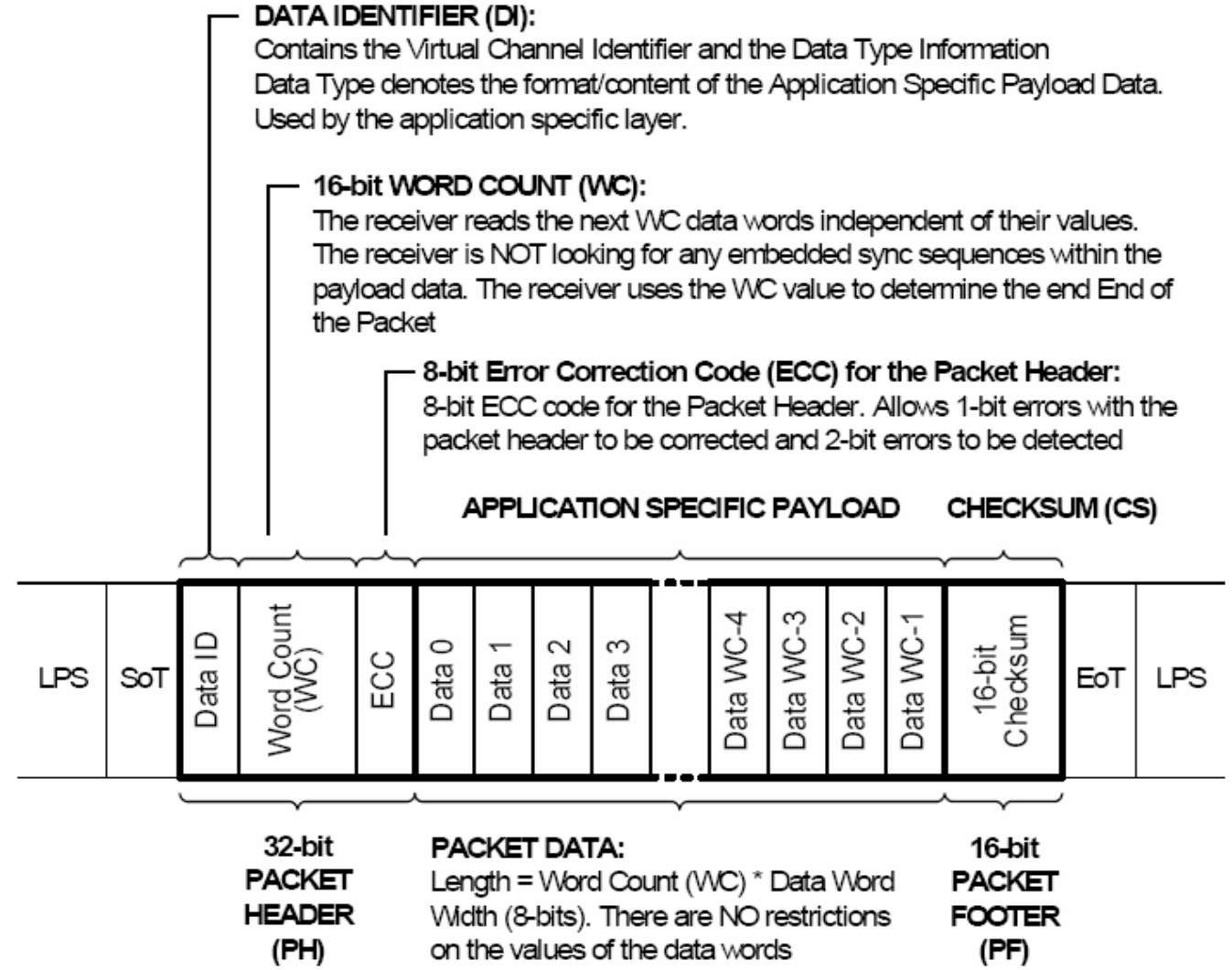
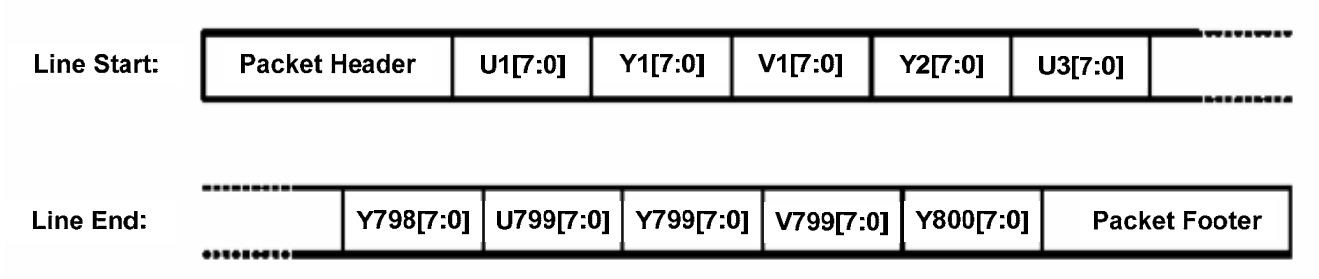


Figure 14: Long Packet Structure



[NOTE] Byte values transmitted LSB first.

Figure 15: YUV422 Transmission

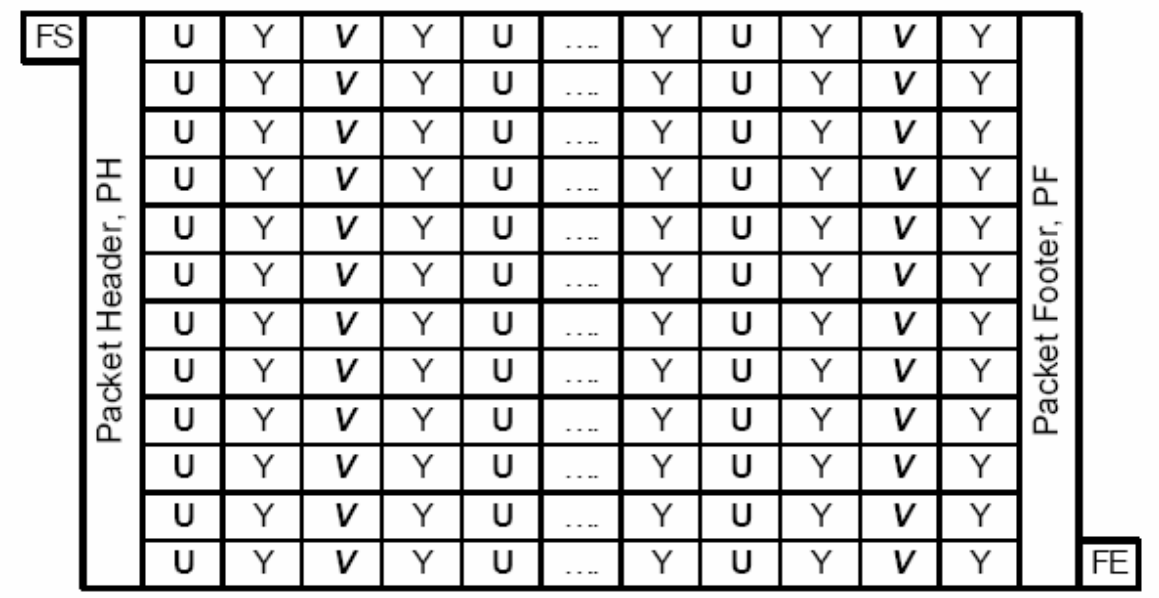
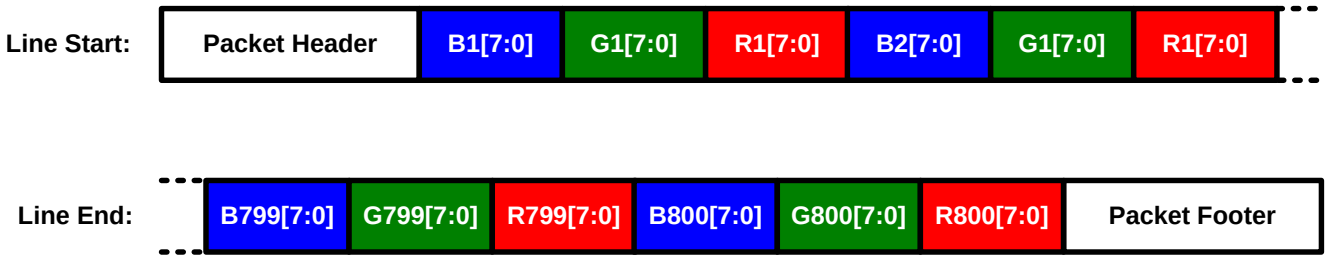


Figure 16: YUV422 Frame Format



[NOTE] Byte values transmitted LSB first.

Figure 17: RGB888 Transmission

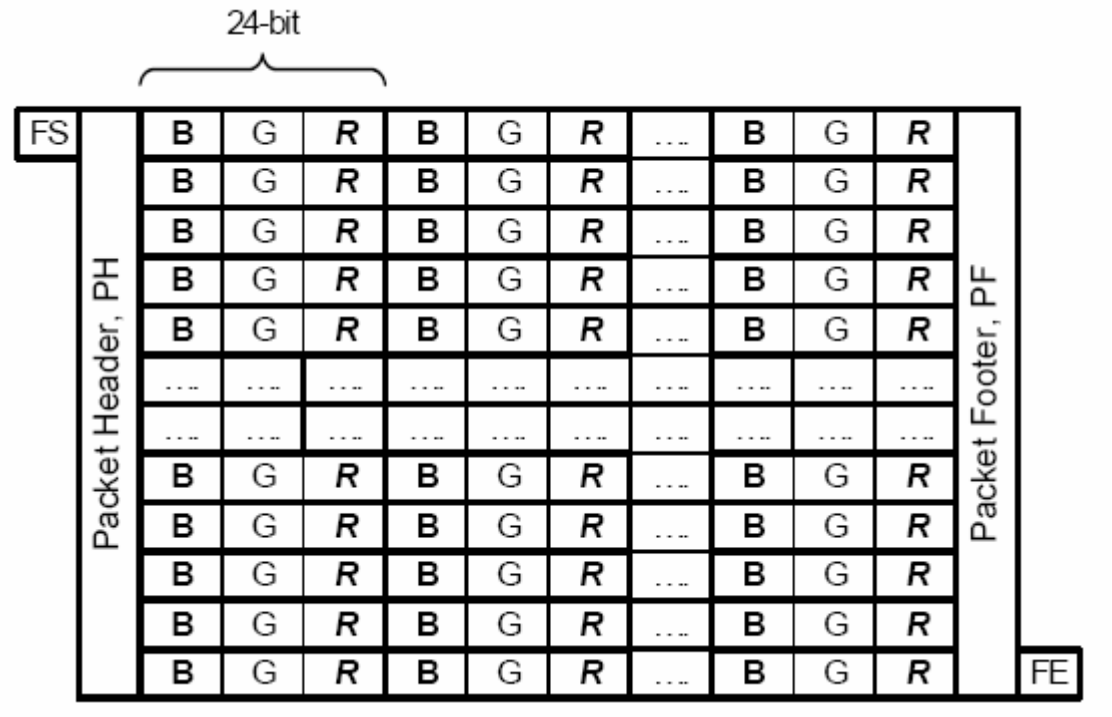


Figure 18: RGB888 Frame Format

Line Start:	**Packet Header**	**BGR1[15:0]**	**BGR2[15:0]**	**BGR3[15:0]**
Line End:	**BGR798[15:0]**	**BGR799[15:0]**	**BGR800[15:0]**	**Packet Footer**

[NOTE] Byte values transmitted LSB first.

Figure 19: RGB565 Transmission

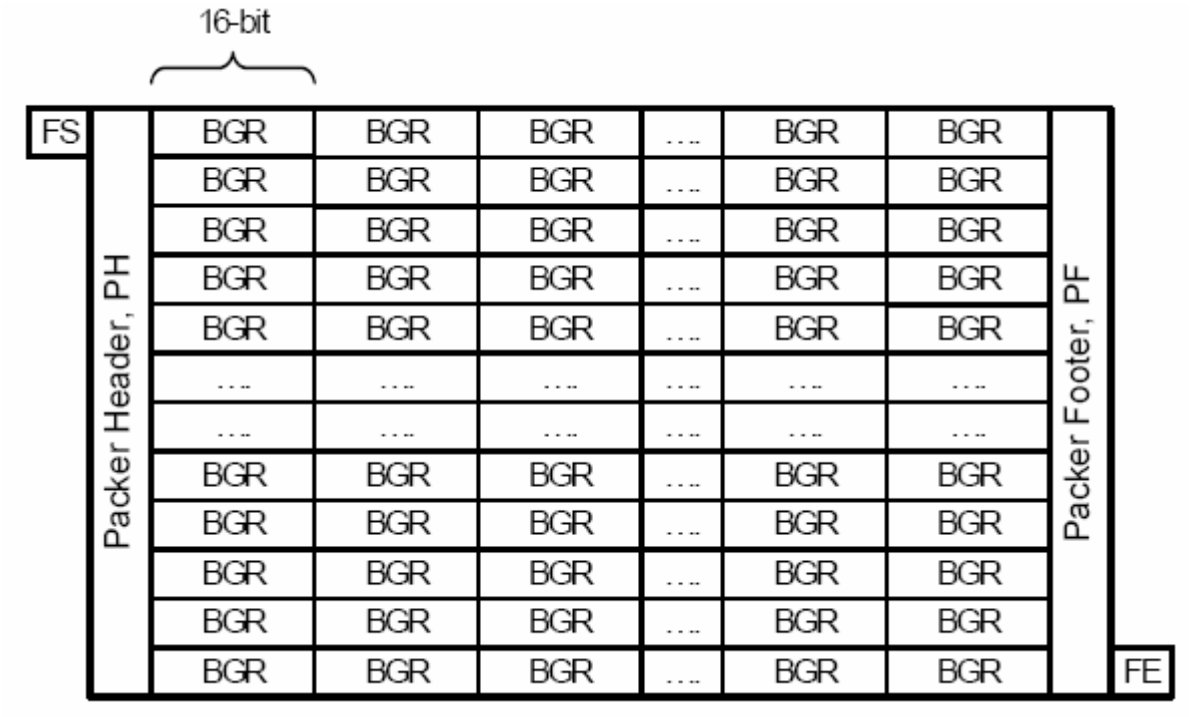
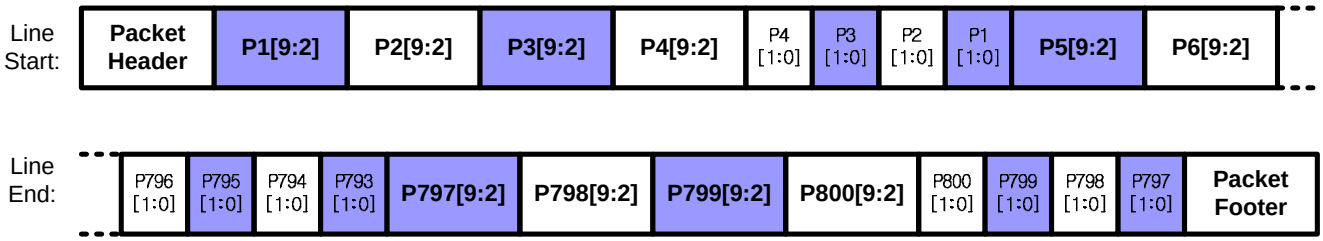


Figure 20: RGB566 Frame Format



[NOTE] Byte values transmitted LSB first.

Figure 21: RAW10 Transmission

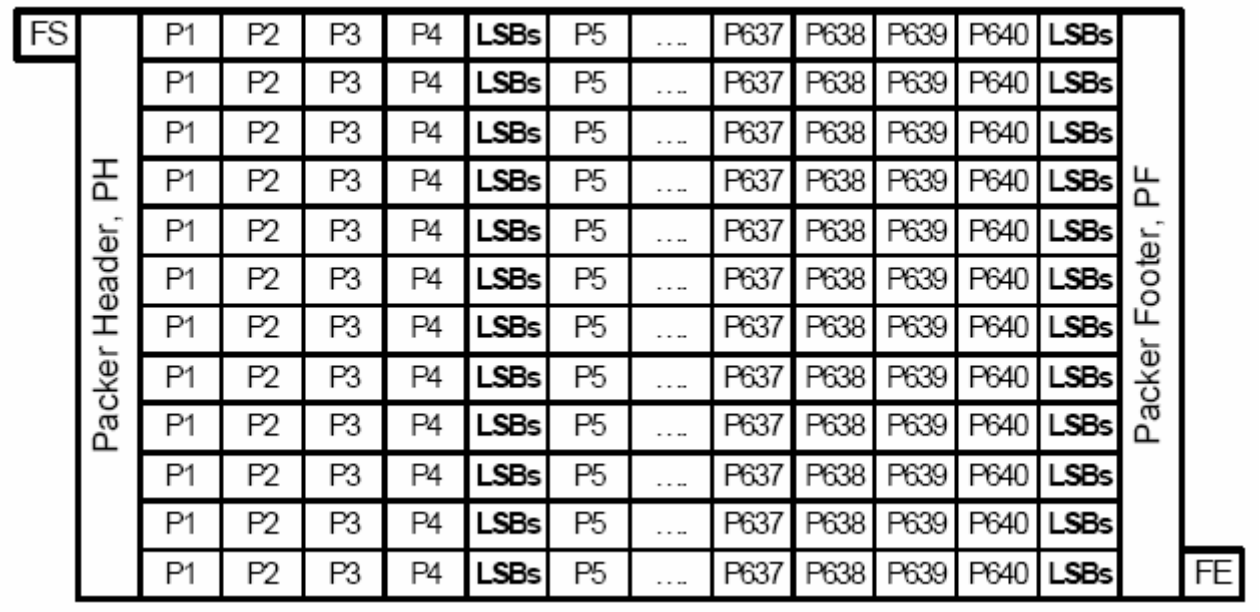


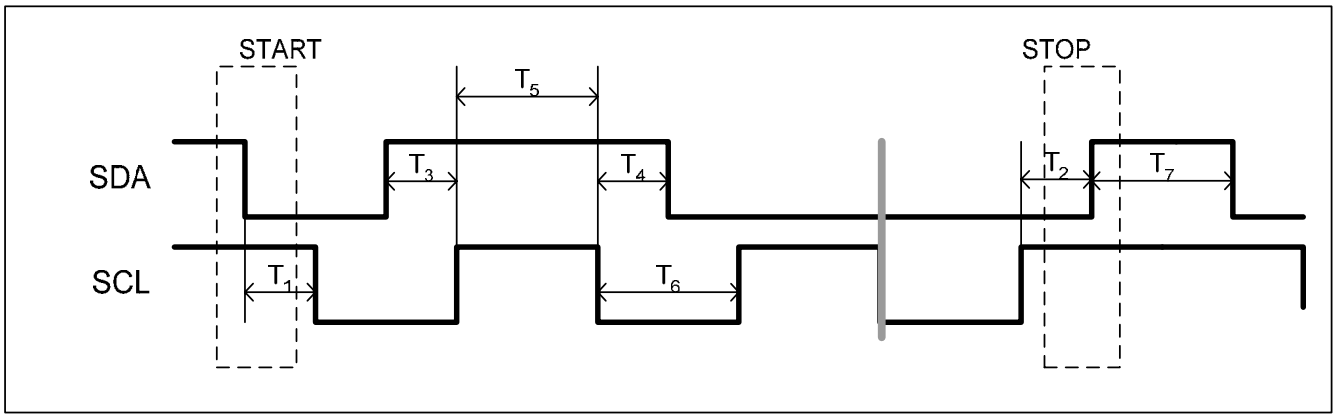
Figure 22: RAW10 Frame Format

CONTROL INTERFACE DESCRIPTION

The I^2C interface is a two-wire bi-directional serial bus. Both wires (Serial Clock Line -SCL and Serial Data Line - SDA) are connected to a positive supply via a pull-up resistor, and when the bus is free both lines are high. The output stage of the device must have an open-drain or open collector type IO cell so that a wired-AND function between all devices that are connected on the bus can be performed.
The two-wire serial interface defines several different transmission stages, as follows:

- A start bit
- The slave device 7-bit address
- An (No) acknowledge bit coming from slave.
- An 8-bit or 16-bit message (address and/or data).
- A stop bit (or another 8bit or 16bit message in multiple Read/Write access)

The data on the SDA pin must be stable during the high period of the clock (SCL) as shown in the figure below. Only the master may change the data while SCL is high. A high-to-low transition marks a START condition, and a low-to-high a STOP condition.



SYMBOL	PARAMETER	MIN	MAX	UNIT
	SCL clock frequency	0	100	kHz
T1	Hold time for START condition	0.4	-	us
T2	Setup time for STOP condition	4.0	-	us
T3	Data setup time	250	-	ns
T4	Data hold time	0	3.45	us
T5	High period of the SCL clock	4.0	-	us
T6	Low period of the SCL clock	4.7	-	us
T7	Bus free time between STOP and START condition	4.7	-	us
	Rise time for both SDA and SCL signals		1000	ns
	Fall time for both SDA and SCL signals		300	ns
C_B	Capacitive load for each bus line		400	pF

(a) Standard Mode

SYMBOL	PARAMETER	MIN	MAX	UNIT
	SCL clock frequency	0	400	kHz
T1	Hold time for START condition	0.6	-	us
T2	Setup time for STOP condition	0.6	-	us

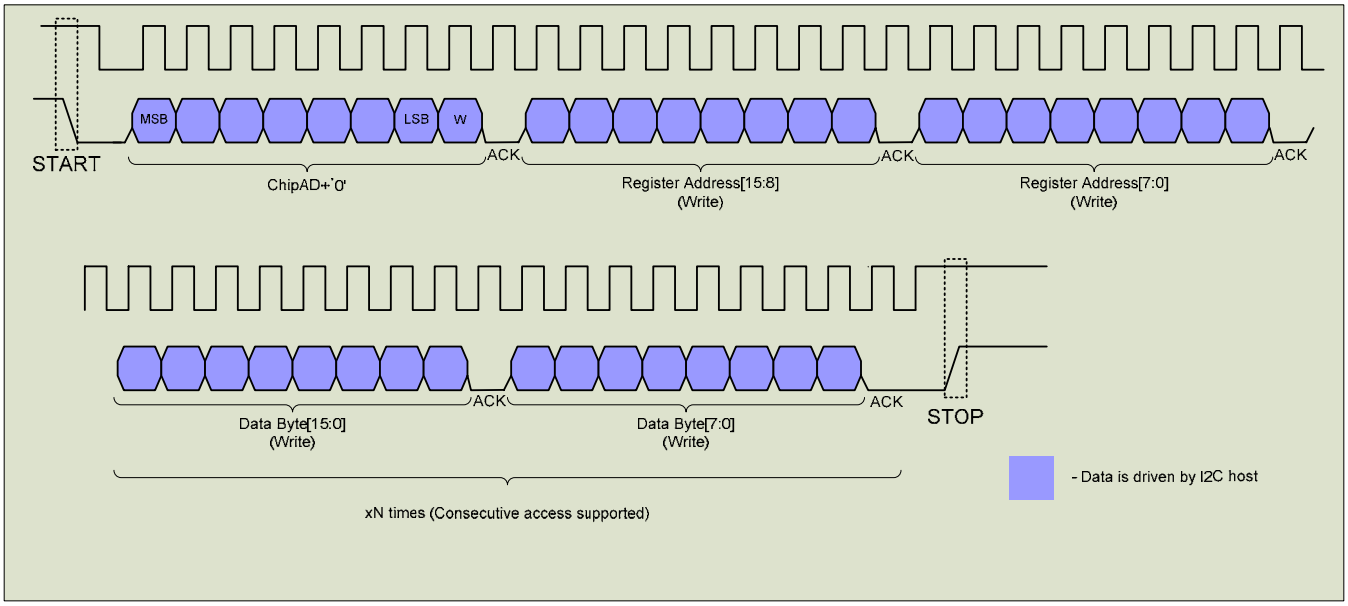
T3	Data setup time	100	-	ns
T4	Data hold time	0	0.9	us
T5	High period of the SCL clock	0.6	-	us
T6	Low period of the SCL clock	1.3	-	us
T7	Bus free time between STOP and START condition	1.3	-	us
	Rise time for both SDA and SCL signals		300	ns
	Fall time for both SDA and SCL signals		300	ns
C_B	Capacitive load for each bus line		400	pF

(b) Fast Mode

Figure 23: I²C General Timing Specification

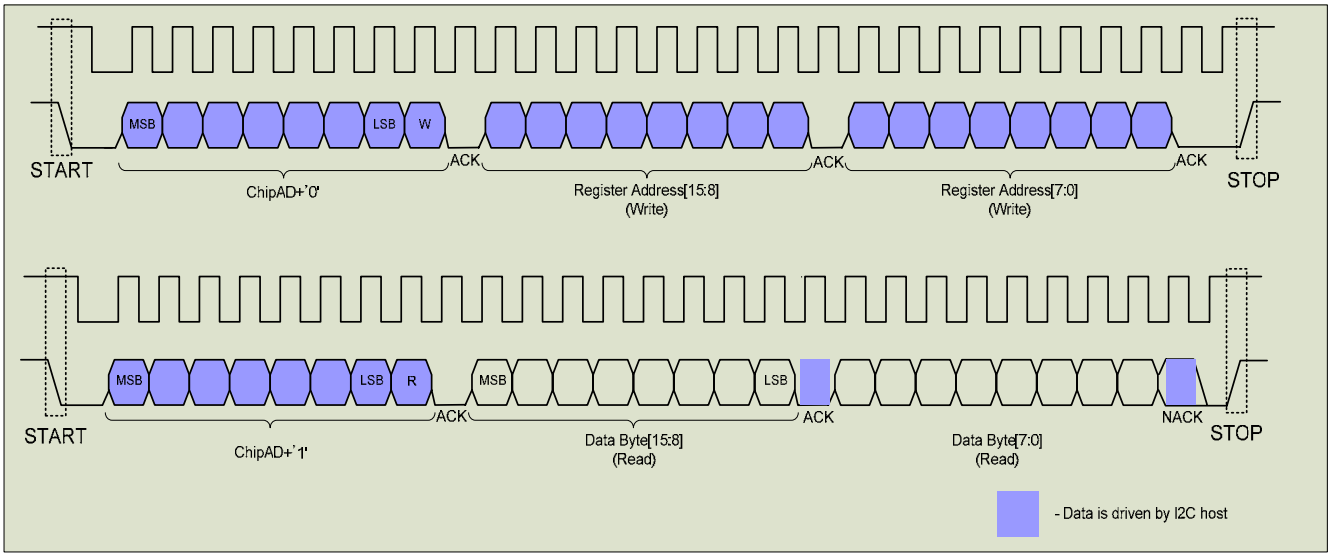
The master device activates a START condition, and sends the first byte of data that contains the 7-bit address, and a direction bit (R/W#, 1 for read, 0 for write). The addressed device answers by pulling down the SDA line as an acknowledge procedure.
Detailed sequences of read and write data transfers are shown in the figures below.

- The colored boxes represent master-to-slave data transfer.
- The clear boxes represent slave-to-master data transfer.



[NOTE] The device address can be changed by pin configuration of IIC_ID, which is described in pad description.

Figure 24: I²C Write Timing Example(16 Address, 2 data bytes)



[NOTE] The device address can be changed by pin configuration of IIC_ID, which is described in pad description.

Figure 25: I²C Single Read Timing Example(16 Address, 2 data bytes)

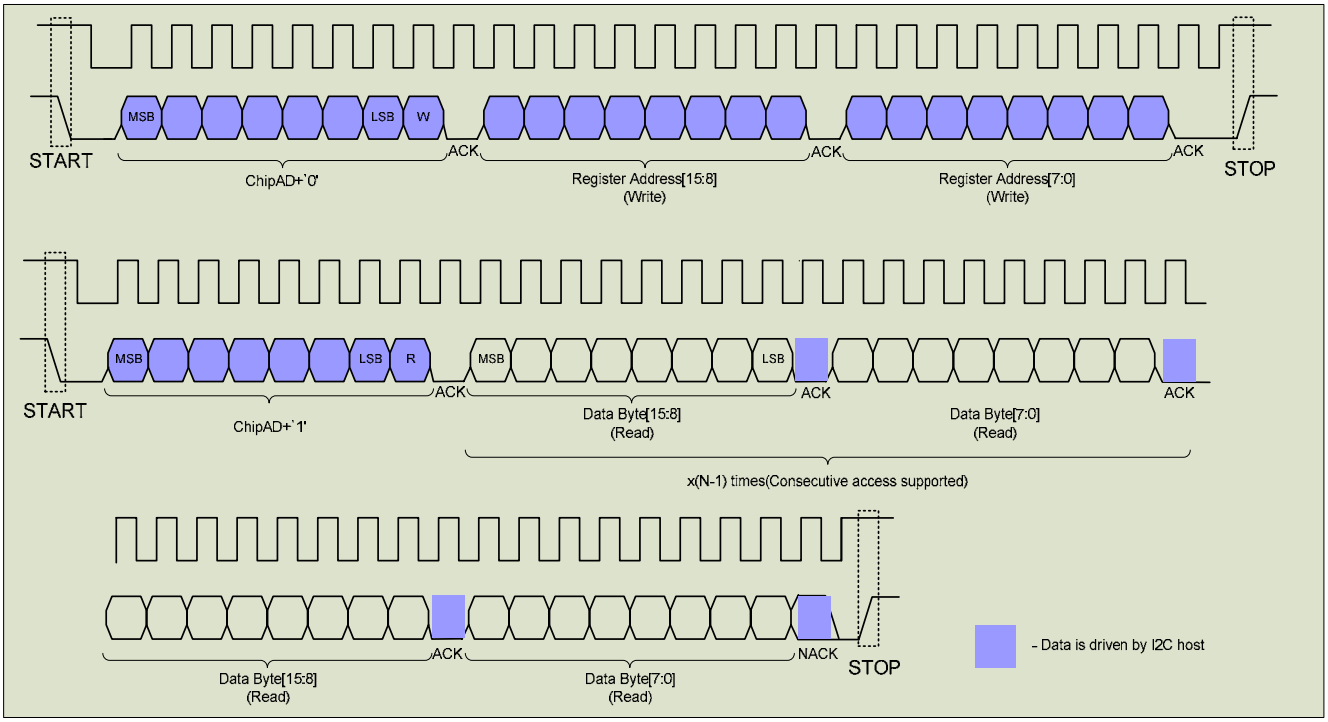


Figure 26: I²C multiple(N) Read Timing Example(16 Address, 2 data bytes)

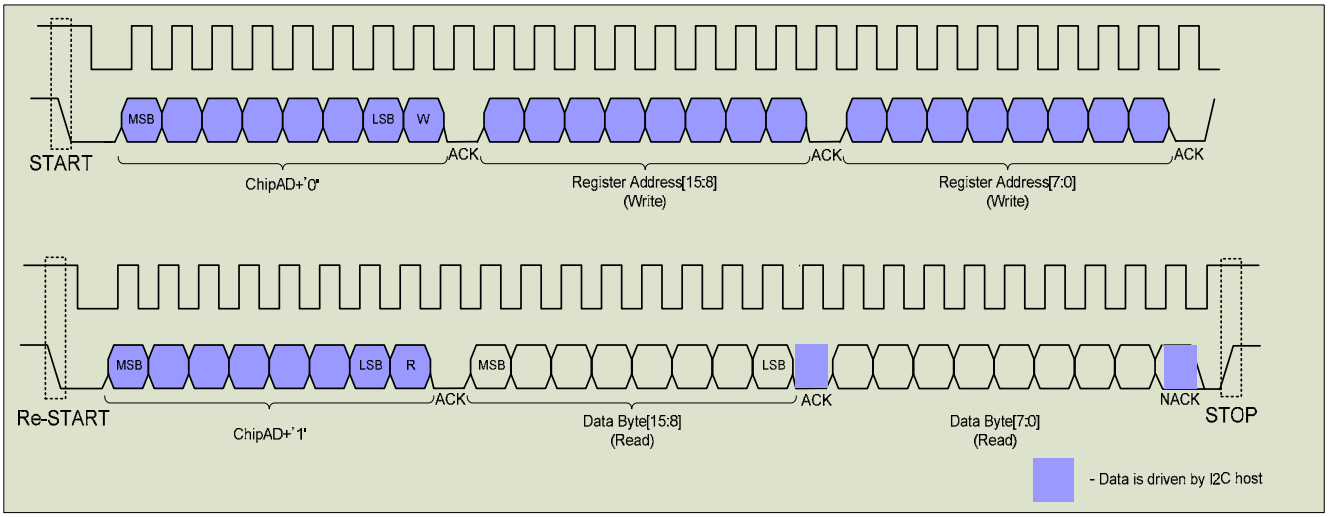
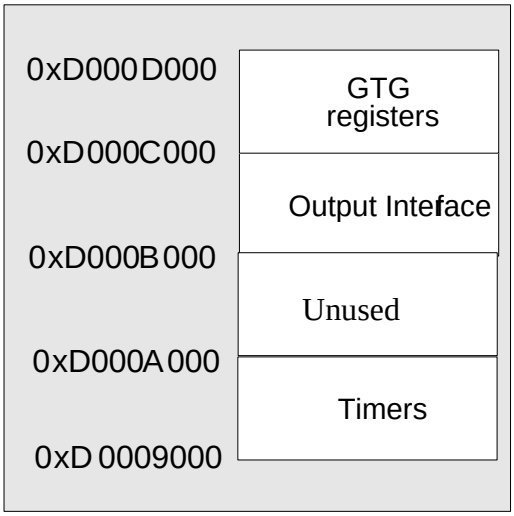


Figure 27: I^2C Single Read acess with Repeated Start Example(16 Address, 2 data bytes)

IIC Writing Example

IIC_ID	Device Address
0	0111_100b/3Ch
1	0101_101b/2Dh

Configure IIC_ID=0, for device address, 0111_100b.



When accessing one of GTG Registers, its page address is C0. You can access it in 8-bit access mode or 16-bit access mode.

ex) When writing data(AAh) to register (04h) of page C0 in 8-bit access mode;
write(78h, FEh, C0h) // set page C0

write(78h, 04h, 00h) // upper byte
write(78h, 05h, AAh) // lower byte
[NOTE] write(device address & R/W bit, register address, data, ...)
[NOTE] All data are regarded as 16-bits.

ex) When writing a series of data to continuous registers of page (C0h) in 8-bit access mode;
data(AAh) -> register(04h)
data(BBh) -> register(06h)
data(CCh) -> register(08h)
data(DDh) -> register(0Ah)
write(78h, FEh, C0h) // set page C0
write(78h, 04h, 00h, AAh, 00h, BBh, 00h, CCh, 00h, DDh)

ex) When writing a series of data to continuous registers of page (C0h) in 16-bit access mode;
write(78h, C0h, 04h, 00h, AAh, 00h, BBh, 00h, CCh, 00h, DDh)

FUNCTIONAL DESCRIPTION

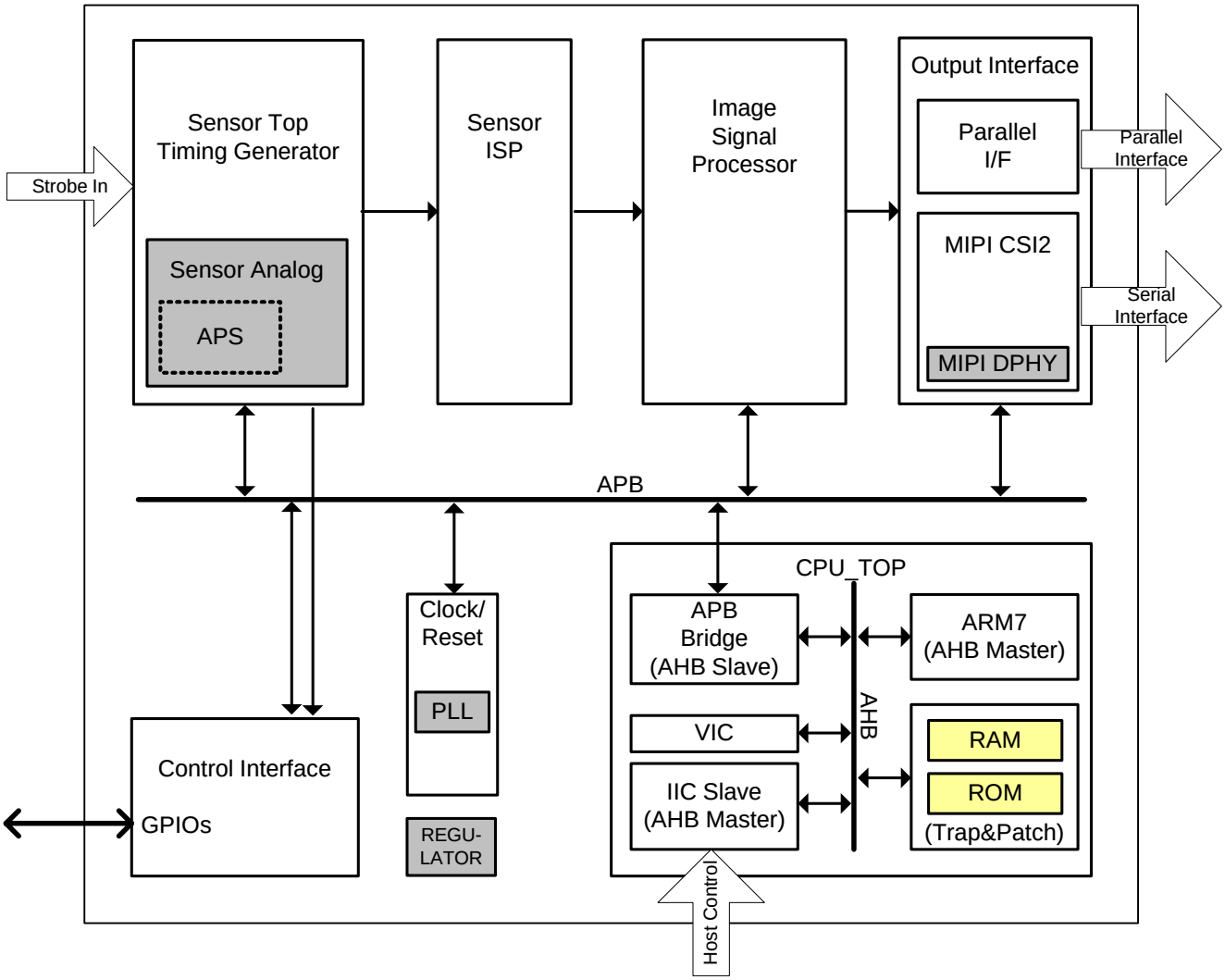


Figure 28: Functional Block Diagram

1. Analog to Digital Converter (ADC)

The image sensor has an on-chip ADC. Column parallel ADC scheme is used for low power analog processing.

1-1. Correlated Double Sampling (CDS)

The analog output signal of each pixel includes some temporal random noise caused by the pixel reset action and some fixed pattern noise by the in-pixel amplifier offset deviation. To eliminate those noise components, a correlated double sampling (CDS) circuit is used before converting to digital code. The input signal level of each pixel is determined as the differential value between the pre-reset pixel value and its current charged one. Therefore its value is sampled twice during a pixel period, once for the reference(reset) level detection and then

the actual signal level.

1-2. Programmable Gain

The user can control the gain of pixel signal by Gain Control Register. As increasing the signal gain control register, the ADC conversion range slope becomes decreased and its output code value is increased. The gain increased as following equation:

Analog Gain

register gain[7:0]

$$(\text{Analogue gain}) = 2^{GAIN[7:5]} \times \left(1 + \frac{GAIN[4:0]}{32}\right)$$

Figure 29: Relative Channel Gain

1-2. PROGRAMMABLE OFFSET TBD

The user can control the offset of pixel signal by

2. Timing Generator Functions

2-1. CIS RAW DATA OUTPUT

GTG supports configurable-bit CIS raw data.

PIXEL ARRAY ADDRESSES

An addressable pixel array is defined as the pixel address range to be read. The addressable pixel array can be assigned anywhere on the pixel array. The addressed region of the pixel array is controlled by ***x_addr_start***, y_***addr_start***, ***x_addr_end*** and y_***addr_end*** register.

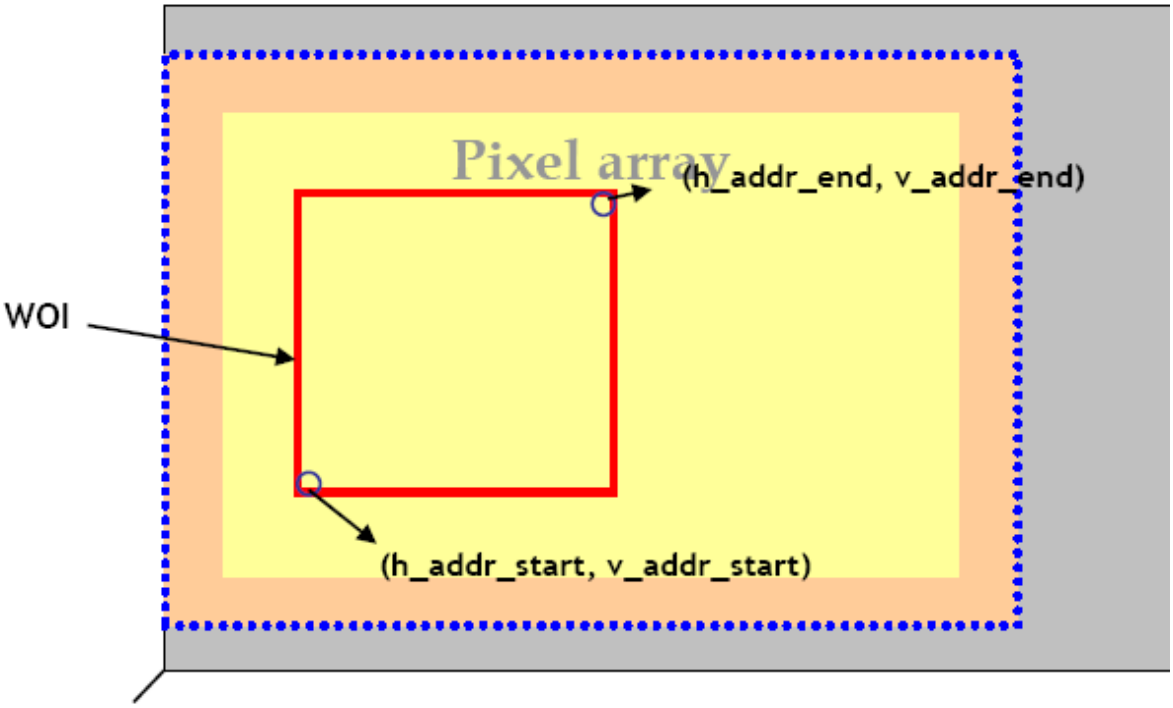


Figure 30: Window of interest of Pixel Array

MIRROR/FLIP

The pixel data is normally read out from left to right in the horizontal direction and from top to bottom in the vertical direction. By changing the ***mirror/flip*** mode, the read-out sequence can be reversed, and the resulting image can be flipped like a mirror image. Pixel data is then read out from right to left in horizontal mirror mode and from bottom to top in vertical flip mode. The horizontal mirror and the vertical flip mode can be programmed by the image orientation register.

The sensor module supports four possible pixel readout orders, as described in the sections below.

STANDARD READOUT

The addressed region of the horizontal pixel data output is controlled by the ***x_addr_start, x_addr_end*** register, and the addressed region of the vertical pixel data output is controlled by the ***y_addr_start, y_addr_end*** register.

HORIZONTALLY MIRRORED AND VERTICALLY FLIPPED READOUT

The addressed region of the horizontal pixel data output is controlled by the ***x_addr_end, x_add_start*** register, and that of the vertical pixel data output is controlled by the ***y_addr_end, y_add_start*** register.

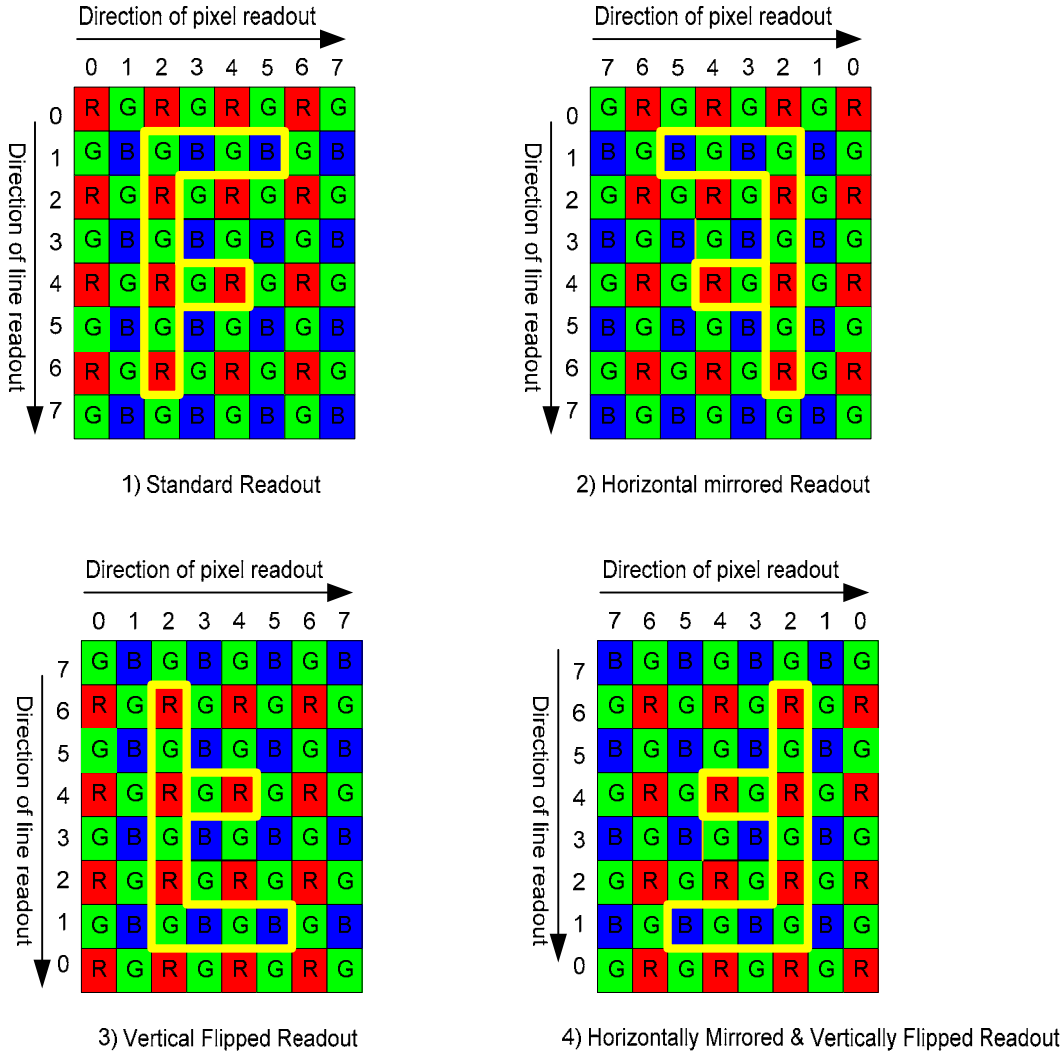


Figure 31: Horizontal Mirror and Vertical Flip

SUB-SAMPLED READOUT

By programming the x and y odd and even increment registers (x_even_inc, x_odd_inc, y_even_inc, y_odd_inc), the sensor can be configured to read out sub-sampled pixel data.

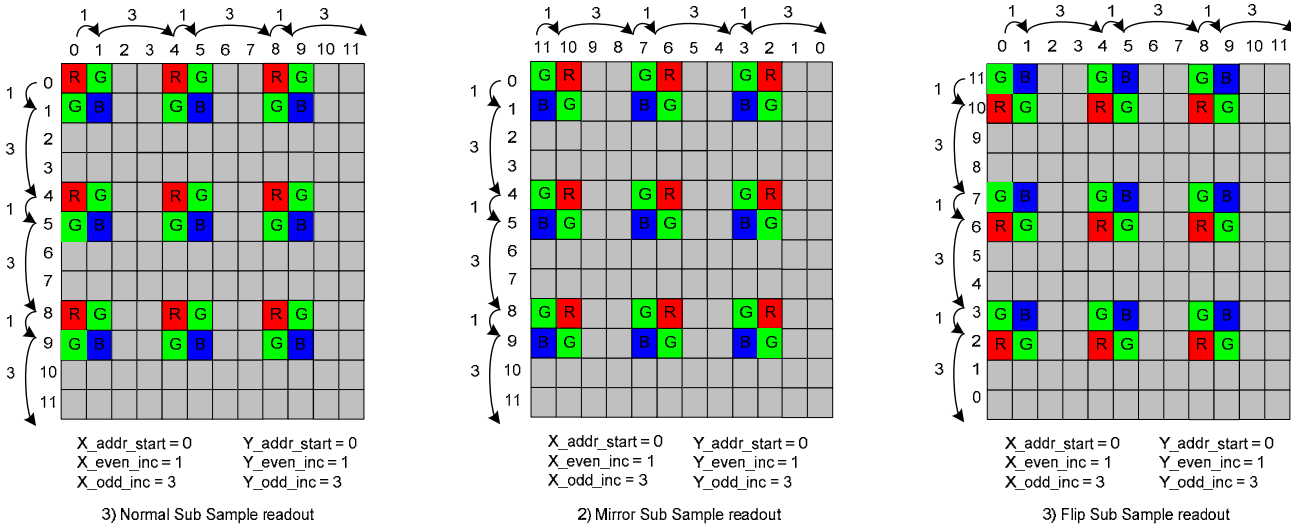


Figure 32: Sub-sampled readout

☞ **NOTE: All figure examples are related to the red first array structure. Generic TG also supports green first array structures.**

FRAME RATE CONTROL (VIRTUAL FRAME)

The line rate and the frame rate can be changed by varying the size of the virtual frame. The virtual frame’s width and depth are controlled by the line_length_pck and frame_length_lines register. The horizontal and vertical blanking times (horizontal blanking time: **line_length_pck** – ***x_output_size***, vertical blanking time: **frame_length_lines** – ***y_output_size***) should meet system requirements.

Frame rate = TGCLK / (frame_length_lines * line_length_pck)

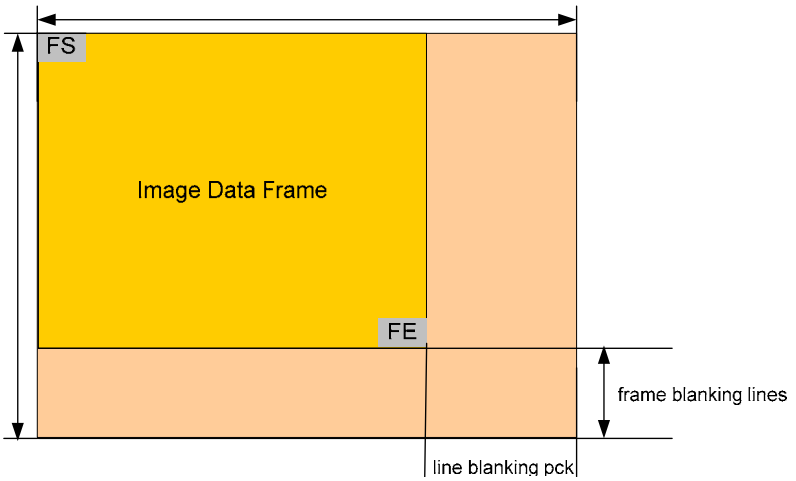


Figure 33: Virtual Frame Timing

INTEGRATION TIME CONTROL (ELECTRONIC SHUTTER CONTROL)

The pixel integration time is controlled by the shutter operation. During the shutter operation, the amount of time – integration time – is determined by the column Step Integration Time Control Register (fine_integration_time) and the line Step Integration Time Control Register (coarse_integration_time). The total integration time of the sensor module can be calculated using the following formula:

Total_integration_time = {(coarse_integration_time * line_length_pck) + fine_integration_time + const} * pclk period [sec]

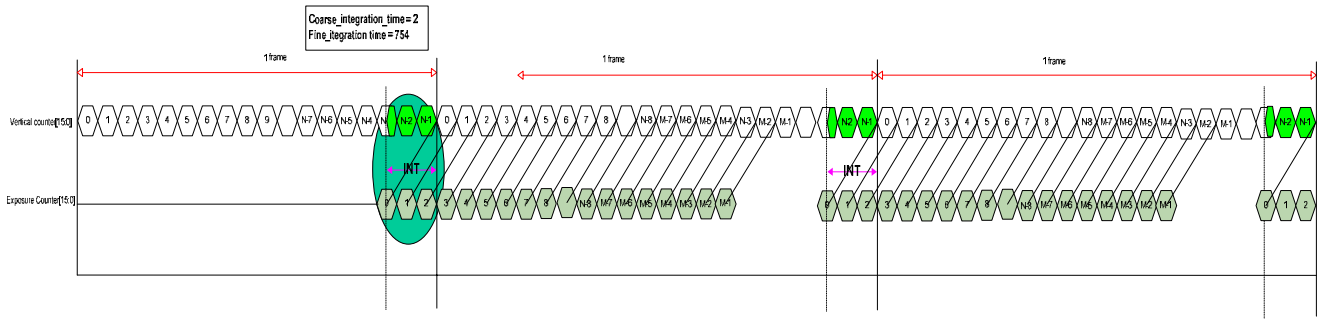


Figure 34: Integration time counter diagram

2-8. LED Flash Control

Both devices are controlled by the firmware and activated directly by the sensor.

Register-Based Host Interface

The following registers control the flash status and functionality:

REG_TC_FLS_Mode	Sets flash mode according to TC_FlashSt_type enum
REG_TC_FLS_Threshold	Sets flash activation threshold in normalized brightness units
REG_TC_FLS_Polarity	Sets flash device polarity. 1: active high, 0: active low

[NOTE]

- There is no guarantee for the quality of AE or any other algorithm convergence before the flash capture. There is typically only one frame for convergence. This time frame is too short, and the results may not be perfect.
- Using an extra frame for AE or AWB convergence extends the preview to capture time. Typically, it doubles this period.

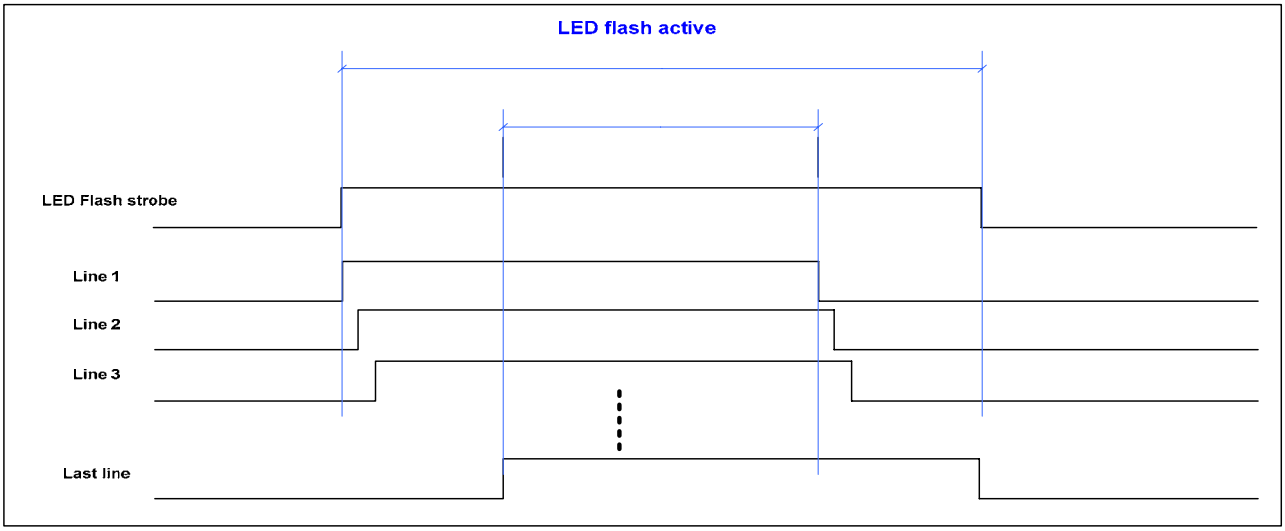


Figure 35: Flash Timing Diagram

Following are application examples for each flash device type. Please note that when LED is used, the host is responsible for algorithm convergence prior to capture, the FW is responsible for algorithm convergence. Dedicated flash algorithms convergence code can be added to the FW using a special SW hook function that is loaded to the FW during initialization.

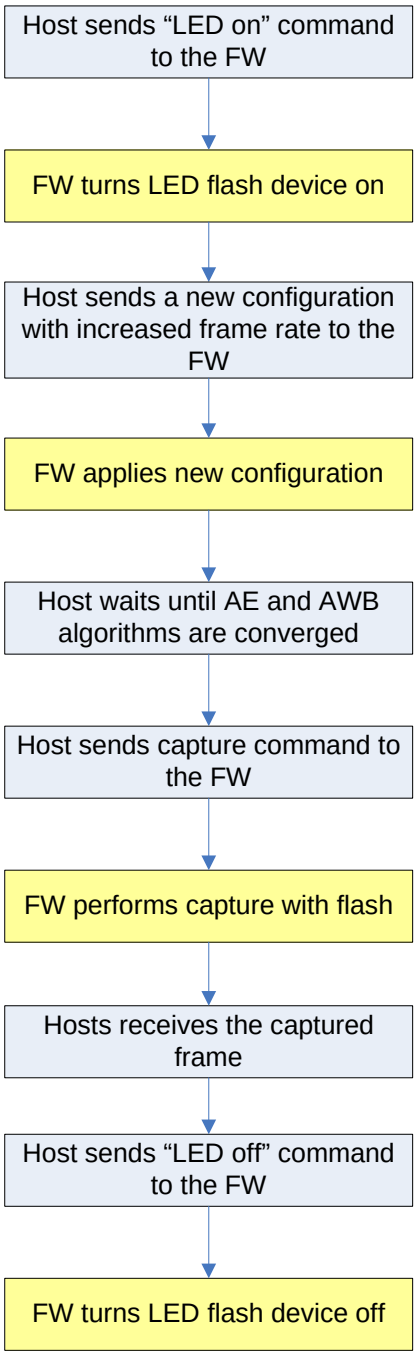


Figure 36: LED Flash Capture Sequence

3. Image Signal Processor

3-1. Auto Exposure

The embedded AE control algorithm tracks the change of the luminance in selected windows, and then compares it to the AE target value. The target value varies according to the scene type. The image brightness is adjusted by controlling analog and digital gains and image sensor integration time. The AE algorithm is designed for fast convergence and may adapt very quickly to dynamic illumination changes.

3-2. Auto White Balance

The AWB algorithm alters the color components of the image in order to ensure that the white color appears white under all illumination types. The algorithm uses three different statistics channels – one per illumination group (warm, outdoor and general). Each channel filters the image pixels based on R-gain / B-gain plane polygon.
The algorithm includes a scene type detector (six scene types).

3-3. Auto Flicker Correction

Flicker may occur when the sensor integration time is not an integer multiple of the frequency of electrical network, for example under a 50Hz or 60Hz fluorescent lamp. The flicker is detected using a dynamic algorithm and can be corrected by adjusting the integration time to some limited values. If the exposure value is smaller than 1/100 of a second (or 1/120, depending on the lighting frequency), flicker band noise may be seen in an office environment.

3-4. Lens Shading Correction

Two different methods of shading correction are used– one uses parabolic shading compensation, and another removes residual effects and is based on grid model. Shading correction dynamically changes based on illumination type.

3-5. Color Demosaicking

Each Bayer color pixel from the image sensor is converted into an RGB pixel and the missing color information of a Bayer pixel is derived from the value of adjacent pixels. The algorithm uses several special-purpose approaches such as text and natural modes. Separate decisions are made for each pixel in the image.

3-6. Color Correction

Variable color profiles are used for color representation improvement. The decision about the profile is taken based on scene brightness and illumination type. Color correction is done using non-linear transformation, parameterized by 18 coefficients, based on ICC device-link technology.

3-7. Despeckle

This algorithm detects and replaces isolated bad pixels and pixel pairs on the raw image data based on their neighbors’ pattern and average.

3-8. Denoising

The denoising algorithm implements the "edge-preserving smoothing" algorithm. It averages pixels that are close in value to the central pixel. Neighboring pixels are equalized before averaging.

3-9. Gamma Correction

Five Gamma correction tables are used for the following color components:
R, G and B – Contrast and device correction
Y – Luminance correction
UV – For color saturation correction

3-10. Image Downscaling

The image from the sensor can be downscaled to an arbitrary size with even X and Y dimensions. The downscaling accuracy ensures any output size up to a 3-5 pixel variance. More precise output sizes can be achieved by cropping. Among other resolutions, SXGA, SVGA, VGA, QVGA, QQVGA, CIF, and QCIF resolutions

are supported. In order to increase the frame rate, averaged sub-sampled scaling is supported for output sizes of VGA and below.

3-11. Special Effects

The special effects may be used to create a Sepia (warm tone), Aqua (cool tone), Monochrome or Negative effect on Image.

3-12. Output Formatting

The ISP outputs 8-bit processed video data in the form of standard YUV ITU-R.656/601 or RGB data. Raw sensor data in Bayer format may also be outputted with 8-10 bit accuracy.

3-13. Image Properties Controls

The user may dynamically control the following image properties independently - Brightness, Contrast, Saturation, Sharpness, Glamour.

SYSTEM STATE DIAGRAM

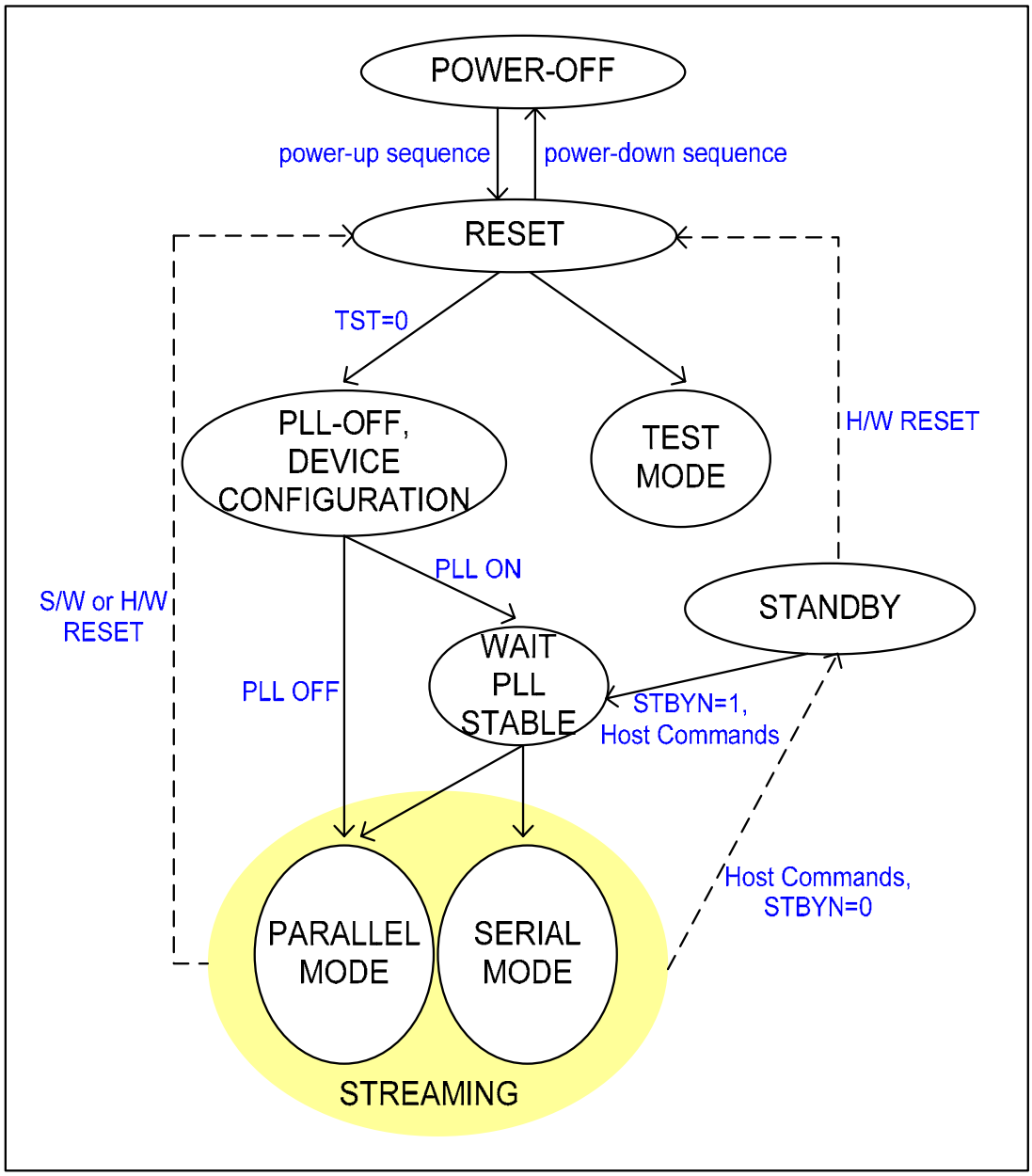
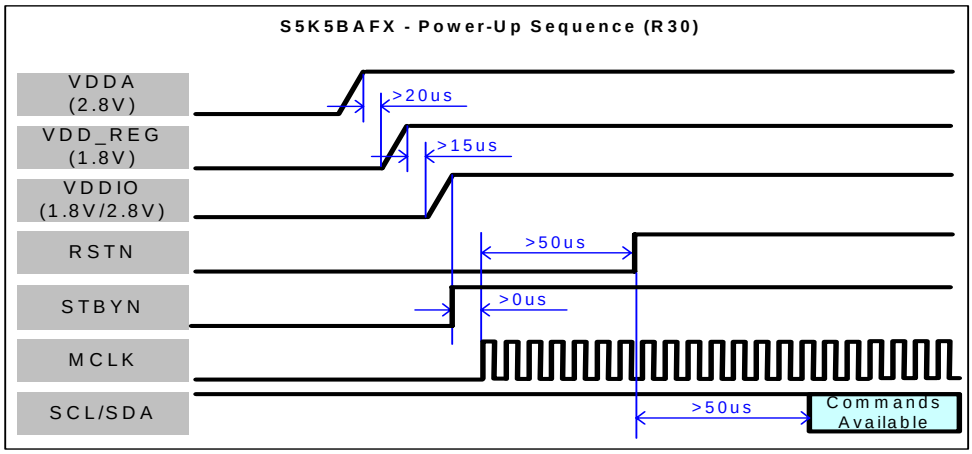


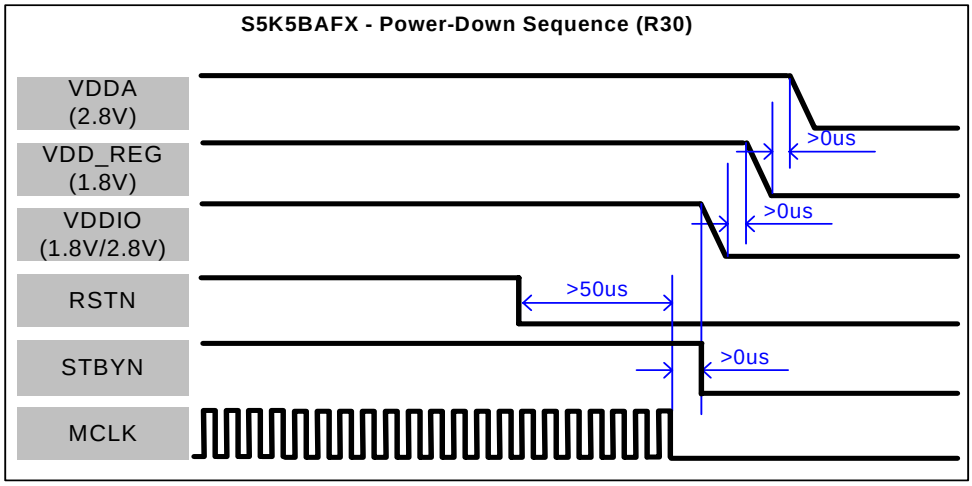
Figure 37: System State Diagram

POWER-UP/DOWN SEQUENCE



[NOTE] If internal regulator is not used, open VDD_REG and apply VDD15, of which power-up sequence is same to VDD_REG.

Figure 38: Power-Up Sequence



[NOTE] If internal regulator is not used, open VDD_REG and apply VDD15, of which power-down sequence is same to VDD_REG.

Figure 39: Power-Down Sequence

STANDBY SEQUENCE

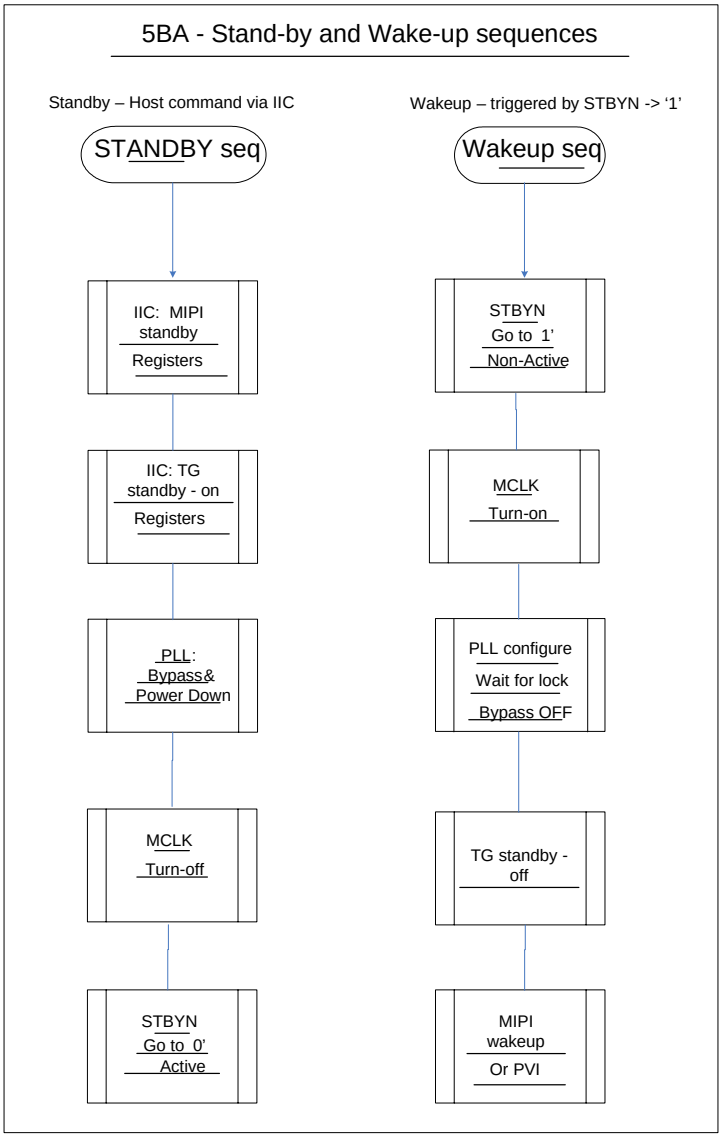


FIGURE 40: STANDBY/WAKEUP- SEQUENCES

ELECTRICAL CHARACTERISTICS

Table 2: Absolute Maximum Rating

Parameter	Symbol	Value	Unit
I/O Digital Power (2.8V or 1.8V)	V_{DDIO}	-0.3 to 3.8	V
Analog Power (2.8V)	V_{DDA}	-0.3 to 3.8	
Core Digital Power (1.5V)	V_{DDD}	-0.3 to 2.0	
Input Voltage	V_I	-0.3 to 3.8	
Ambient Temperature	T_A	-20 to +60	°C
Storage Temperature	T_S	-40 to +85	

Table 3: DC Characteristics

(V_{DDIO1} = 2.8V ± 0.2V, V_{DDIO2} = 1.8V ± 0.10V , V_{DDD} = 1.5V ± 0.1V, Ta = -20 to + 60 °C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage	V_{DDA}		2.6	2.8	3.0	V
	V_{DDD}		1.40	1.5	1.60	
	V_{DDIO1}		2.6	2.8	3.0	
	V_{DDIO2}		1.7	1.8	1.90	
High-Level Input Voltage	V_{IH}		0.7* V_{DDIO}	-	-	
Low-Level Input Voltage	V_{IL}		-	-	0.2* V_{DDIO}	
High Level Output Voltage	V_{OH}		V_{DDIO}-0.2	-	-	
Low-Level Output Voltage	V_{OL}		-	-	0.2	
High-Level Input Current	I_{IH}	$V_I = V_{DDIO}$	-10	-	10	uA
		$V_I = V_{DDIO}$ (with Pull-Down)	-	-	72	
Low-Level Input Current	I_{IL}	$V_I = V_{SS}$	-10	-	10	
		$V_I = V_{SS}$ (with Pull-Up)	-72	-	-	
Standby Current	I_{STBY}	STBYN = Low, MCLK = Low (0 lux Illumination)	-	200	250	
Supply Current	I_{DD}	Serial Output Mode @15fps	81	95	180	mA
		Parallel Output Mode @15fps	TBD	-	TBD	
Power Consumption	P_{DD}	Serial Output Mode @15fps	120	-	270	mW
		Parallel Output Mode @15fps	TBD	-	TBD	
Input Capacitance	C_{IN}		-	-	TBD	pF

IMAGING CHARACTERISTICS

Table 4: Imaging Characteristics

Parameter	Unit	Value	Remark
Effective resolution	pixel	1600 x 1200	
Active resolution	pixel	1616 x 1216	
Optical fomat	Inch	1/5	
Pixel size	Um	1.75um	
Shutter type	-	Electronic rolling shutter	
Full saturation	mV	**TBD**	
ADC saturation	mV	**TBD**	
Sensitivity	mV/lux.sec	**TBD**	
Dark current	mV/sec	**TBD**	
Random noise	e-	**TBD**	
Dynamic range	dB	**TBD**	
Max. SNR	dB	**TBD**	
Max. Gr/Gb ratio		**TBD**	
Max. frame rate	fps	15fps@full resolution 30fps@VGA	
Max. CRA	degree	**TBD**	
ADC resolution	bit	10	

REGISTER DESCRIPTION

ADDRESS MEMORY MAP

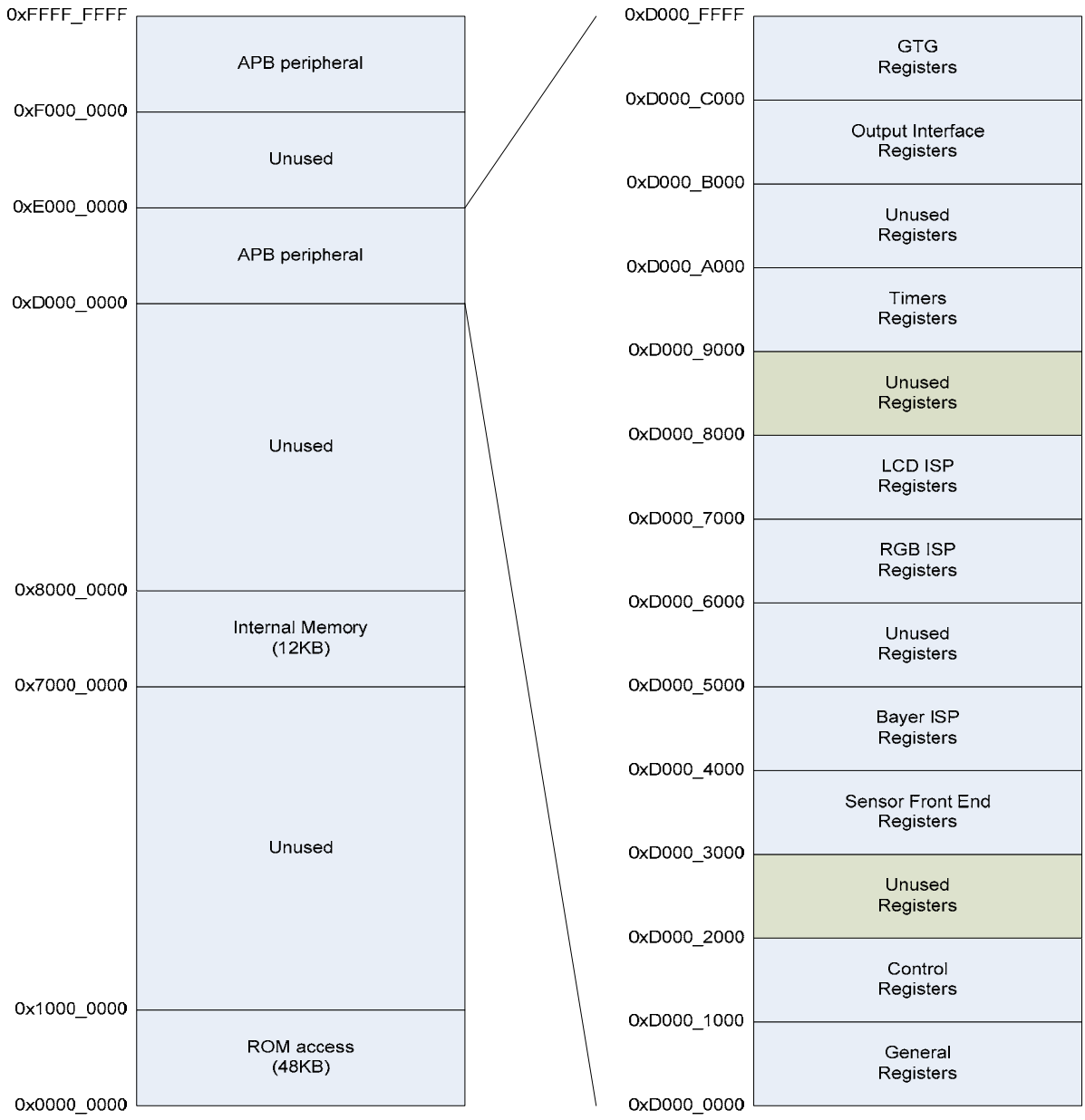


그림 1

Peripheral		Memory(Registers) Address
APB Peripheral		0xF000_0000 ~ 0xFFFF_FFFF
Reserved		0xE000_0000 ~ 0xEFFF_FFFF
APB Peripheral	GTG Registers	0xD000_C000 ~ 0xDFFF_FFFF
	Output Interface Registers	0xD000_B000 ~ 0xD000_BFFF
	TG Registers	0xD000_A000 ~ 0xD000_AFFF
	Timers Registers	0xD000_9000 ~ 0xD000_9FFF
	Reserved	0xD000_8000 ~ 0xD000_8FFF
	LCD ISP Registers	0xD000_7000 ~ 0xD000_7FFF
	RGB ISP Registers	0xD000_6000 ~ 0xD000_6FFF
	Reserved	0xD000_5000 ~ 0xD000_5FFF
	Bayer ISP Registers	0xD000_4000 ~ 0xD000_4FFF
	Sensor Front End Registers	0xD000_3000 ~ 0xD000_3FFF
	Reserved	0xD000_2000 ~ 0xD000_2FFF
	Control Registers	0xD000_1000 ~ 0xD000_1FFF
	General Registers	0xD000_0000 ~ 0xD000_0FFF
Reserved		0x8000_0000 ~ 0xCFFF_FFFF
Internal SRAM		0x7000_0000 ~ 0x7000_2FFF(12KB)
Reserved		0x1000_0000 ~ 0x6FFF_FFFF
Internal Rom		0x0000_0000 ~ 0x0000_BFFF(48KB)

GENERAL REGISTERS(0XD000_0000 ~ 0XD000_0FFF)

I2C_mode (Read/Write) 0xD000_0000

Bits	7	6	5	4	3	2	1	0
Default	0x11							
description	I2C_8bit_add_det_mode	I2C_sync_mode		I2C_lpf			I2c_clock_stretch_dis	I2C_msbfirst

I2C_8bit_add_det_mode - (fcfc_enable) – Select the detect mode for 8bit address access

0 – Switching to 8bit address mode when ‘xxfe’ or ‘fexx’ sequence, detected in address field of I2C transaction.

1 – Switching to 8bit address mode when only ‘fefe’ sequence, detected in address field of I2C transaction. This mode allows using RAM access method via ‘fcfc’.

I2C_msbfirst – Communicate with a data width of 2 bytes. The default is that the MSByte of the I2C transaction (address and data) is received on the interface. The byte order of the data transaction can be changed to LSByte-first by lowering the I2C_msbfirst register to zero.

I2c_clock_stretch_dis –When bit is reset, clock stretching in read cycle is enabled, and last as long as defined in register ***clk_str_delay_reg.***

When this bit is set, it disables the clock stretching on read cycle in Slave I2C. It might be set only if system guarantee for imidiate response since AHB request to AHB response (and readback data arrives no later than **TBD** clock cycles since requested).

I2C_lpf – Use optional low pass filter on the I2C (default is bypass)

I2C_sync_mode – The minimal I2C clock duration is determined by the type of synchronization used in the I2C unit.

00 – Sampling input on each clock – minimum of 8 internal clocks required for the I2C clock.
01 – Sampling input on each second clock – minimum of 12 internal clocks required for the I2C clock.
10 – Sampling input on each third clock – minimum of 18 internal clocks required for the I2C clock
11 – Sampling input on each fourth clock – a minimum of 24 internal clocks required for the I2C clock.

Sampling on each clock – 8:1 minimum relationship (due to a different constraint)
Sampling on each second clock – 12:1 minimum relationship (3 samples in half a waveform * 2 clocks * 2 phases)
Sampling on each third clock – 18:1 minimum relationship (3 samples in half a waveform * 3 clocks * 2 phases)
Sampling on each fourth clock – 24:1 minimum relationship (3 samples in half a waveform * 4 clocks * 2 phases)

Reset_I2C_mode(Read/Write) 0xD000_0002

Bits	7	6	5	4	3	2	1	0
Default	0x00							
description								Reset_I2C_mode

Reset_I2C_mode – Setting this register resets the I2C slave to MSB first and 16bit address/data mode.

I2C_dis_addr_inc(Read/Write) 0xD000_0004

Bits	7	6	5	4	3	2	1	0
Default	0x1							
description								I2C_dis_addr_inc

I2C_dis_addr_inc – This bit must be set for every memory access, in order to prevent the auto increment of the I2C address.

I2c_clk_str_delay_cnt(Read/Write) 0xD000_0006

Bits	8	7	6	5	4	3	2	1	0
Default	0x19								
description	I2c_clk_str_delay_reg								

I2c_clk_str_delay_reg – This 9bit register defines the duration of clock stretching delay in read cycle of slave I2C (in terms of internal clock cycles). In other words, it represents the time between readback data has been set up on I2C data signal, SDIN, to release point of the I2C clock signal, SCLK.

It should be configured to ensure minimum setup time of **250nsec** between SDIN and SCLK signals, as required by I2C BUS SPECIFICATION version 2.1.

Default value is 25 decimal, based on internal clock cycle of 10nsec (to ensure set up time of 250nsec).

I2c_ahb_msb_addr_ptr(Read/Write) 0xD000_0008

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0xd000															
description	I2c_ahb_msb_addr_ptr															

I2c_ahb_msb_addr_ptr - This register contains the 16bit MSB of the 32bit address in accessing the AHB bus.

It can be accessed (read/write) in two ways:

a By special reserved address ***ahb_msb_addr_ptr*** which configured to **0xfcfc.**

b Approaching as a general register, only when this register contains the APB peripherals start address (**0xd000**).

sw_reset (Read/Write) 0xD000_0010

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
description	Not used – must be zero							Sw_reset

Sw_reset – When sw_reset is equal to zero, normal operation of the device is enabled. When the host sets this bit to one, the internal reset signal is activated for seven clk cycles. The device operation is stopped, all units are put into idle and all registers are reset to their default state.

sw_core_reset (Read/Write) 0xD000_0012

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
description	Not used – must be zero							Sw_core_reset

Sw_core_reset – When sw_core_reset is equal to zero, normal operation of the CONT core is enabled. When the host sets this bit to one, the internal reset signal to the CONT core is activated for seven clk cycles.

sw_load_complete (Read/Write) 0xD000_0014

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0*							
description	Not used – must be zero							**Sw_load_complete**

Sw_load_complete – When the host finishes the loading process of the program memory, an access to this register, asserting the Sw_load_complete bit, will disable the reset signal to the CONT.

Command_Wr_addH (Read/Write) 0xD000_0020

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0*							
description					Mem_wr_add[19:16]			

Command_Wr_addL(Read/Write) 0xD000_0022

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0000*															
description	Mem_wr_add[15:0]															

The following register defines the start-address (Byte address) for host READ access to the core memory. The CONT unit increments the address after each memory access by the host.

Mem_Rd_addH(Read/Write) 0xD000_0024

Bits	**7**	**6**	**5**	**4**	**3**	**2**	***1***	**0**
Default	*0x0*							
description					Mem_rdr_add[19:16]			

Mem_Rd_addL(Read/Write) 0xD000_0026

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0000*															
description	Mem_rd_add[15:0]															

The second set of host access registers to main memory follows.
The following register defines the start-address (Byte address) for host Write access to the core memory. The CONT unit increments the address after each memory access by the host.

Command_Wr_addH (Read/Write) 0xD000_0028

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
description					Mem_wr_add[19:16]			

Command_Wr_addL(Read/Write) 0xD000_002A

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Mem_wr_add[15:0]															

The following register defines the start-address (Byte address) for host READ access to the core memory. The CONT unit increments the address after each memory access by the host.

Mem_Rd_addH(Read/Write) 0xD000_002C

Bits	7	6	5	4	3	2	*1*	0
Default	*0x0*							
description					Mem_rdr_add[19:16]			

Mem_Rd_addL(Read/Write) 0xD000_002E

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Mem_rd_add[15:0]															

adr_tg_sw_stby_n (Read/Write) 0xD000_0030

Bits	7	6	5	4	3	2	1	0
Default	**0x06**							
description	reserved					n_stby_sw_reg	n_stby_sw_atop	n_stby_sw_tg

tg_sw_stby_n – Software standby bit (Move clock system to SW-STBY mode). This wire does not connect to TG stand-by bit.
atop_sw_stby_n – Software standby bit for ATOP block
reg_sw_stby_n – Disconnected for future use.

adr_fw_wait (Read/Write) 0xD000_0036

Bits	7	6	5	4	3	2	1	0
Default	**0x00**							
description	reserved							Fw_wait

Fw_wait – When writing ‘1’ to this bit, the clock system enter to ***Firmware Wait*** state as defined in power management table.

adr_force_clk_setting (Read/Write) 0xD000_0038

Bits	7	6	5	4	3	2	1	0
Default	**0xFF**							
description	sclk	sclk_byp	ATOP	sclk_oif	sclk_isp	sclk_cin	sclk_tg	sclk_cpu

Force_sclk_cpu_en_n[0] – When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_cpu.
Force_sclk_tg_en_n[1] – When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_tg.

Force_sclk_cin_en_n[2] – Not functional since CIN does not exist.
Force_sclk_isp_en_n[3] – When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_isp.
Force_sclk_oif_en_n[4] – When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_oif.
Force_ATOP_en_n[5] – When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_atop_x2
Force_sclk_byp_en_n[6] - When ‘1’, it enables to permanently turn-on or turn-off clock signal: sclk_byp.
IMPORTANT: If turn-off, only Hardware-reset will release the chip.
Force_sclk_en_n[7] When ‘1’, it enables to permanently turn-off clock signal: sclk.
IMPORTANT: If turn-off, only Hardware-reset will release the chip.

adr_config_clk_setting (Read/Write) 0xD000_003A

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	**0x00**							
description	Dis_En_n _Sclk	Dis_En_n _Sclk_byp	Dis_En_n_ ATOP	Dis_En_n _Sclk_oif	Dis_En_n _Sclk_isp	Dis_En_n Sclk_cin	Dis_En_n Sclk_tg	Dis_En_n Sclk_cpu

All configuration bits are influence only if its corresponding ***force*** bit from register: ***adr_force_clk_setting*** is set.

Disable_Enable_n_sclk_cpu_en_n[0] – When ‘1’, it turn-off clock signal: sclk_cpu.
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclk_tg_en_n[1] – When ‘1’, it turn-off clock signal: sclk_tg.
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclk_cin_en_n[2] – Not functional since CIN does not exist.
Disable_Enable_n_sclk_isp_en_n[3] – When ‘1’, it turn-off clock signal: sclk_isp.
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclk_oif_en_n[4] – When ‘1’, it turn-off clock signal: sclk_cpu.
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclkx2_ATOP_en_n[5] – When ‘1’ turn-off clock signal: sclk_ATOP_x2.
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclk_byp_en_n[6] - When ‘1’, it turn-off clock signal: sclk_byp.
IMPORTANT: If turn-off, only Hardware-reset will release the chip (if its ***force*** bit is set too).
– When ‘0’, it turn-on clock this clock signal.
Disable_Enable_n_sclk_en_n[6] When ‘1’, it turn-off clock signal: sclk.
When ‘0’, it turn-On this clock signal.
IMPORTANT: If turn-off, only Hardware-reset will release the chip (if its ***force*** bit is set too).

PLL REGISTERS

PLL_en (Read/Write) 0xD000_0050

Bits	7	6	5	4	3	2	1	0
Default	0x00							
description	reserved						PLL_FLTEN	PLL_En

PLL_En – This bit Enables the PLL
PLL_FLTEN – This bit Enables the PLL_FLT

PLL config_m (Read/Write) 0xD000_0052

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0D9															
description	reserved							Pll_m (multiplier)								

Pll_m (multiplier) – (div x1 .. x1024)

Multiplier setting multiplies the comparison frequency to get the VCO frequency.

For the relation between the register and M, see the PLL chapter.

PLL config_p (Read/Write) 0xD000_0054

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x009															
description	reserved										Pll_p (pre divider)					

Pll_p (pre-divider) – (div x1 .. x64)

Pre-divider setting divides the input. The loop comparison frequency is Fref/P. For the relation between the register and P, see the PLL chapter.

pll_reg1 (Read/Write) 0xD000_0056

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default			0x1	0x0	0x1		0x0		0x8				0x8			
description	reserved		resistor select for bias current	Charge pump power-down	Charge pump current control		V-I converter current control		Loop filter Coefficient (R control)				Loop filter Coefficient (C control)			

pll_reg1 - PLL configuration register1.
Register input for general setting for PLL

[13] : ***Internal resistor selection*** - for PLL bias current
(0b : External resistor(default), 1b : Internal resistor)
[12] : Charge pump power-down
(0b : normal operation(default), 1b : power-down)
[11:10] : Charge pump current control
[9:8] : V-I converter current control
[7:4] : Loop filter Coefficient (R control)
[3:0] : Loop filter Coefficient (C control)

pll_lock (Read Only) 0xD000_005C

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
description	reserved															Pll_lock

Pll_lock[0:0] – PLL lock active high indication – directly from PLL.

Dividers and VCO:

PLL_sclk_div (Read/Write) 0xD000_0060

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0008															
Description	reserved							Sclk_dec_disable	reserved	Pll_Sclk-div						

Pll_Sclk-div (divider) – (div x1 .. x64)

Sclock divider configuration. Possible divide ratio: 1,2,4,6…64

Sclk_Dec_dis – decoder disable – When high, the Pll_sclk-div routed directly to PLL.
When low, the Pll sclk-div gets the decoder output.

Sclk_dec_disable	Div ratio	***PLL_sclk_div***	PLL input (SDIV)	Comment
1	xN	N	N	Max is 63 (decimal)
0	x1	1	61	decimal
	x2	2	62	
	x4	4	63	
	x6	6	2	
	xN	N	N-4	
	x64	64	60	

PLL_IFclk_div (Read/Write) 0xD000_0062

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0008															
Description	reserved							IFclk_dec_disable	reserved	Pll_IFclk-div						

Pll_IFclk-div (divider) – (div x1 .. x64)

IFclock divider configuration. Possible divide ratio: 1,2,4,6…64

IFclk_Dec_dis – decoder disable – When high, the Pll_IFclk-div routed directly to PLL.
When low, the Pll ***IF***clk_div gets the decoder output.

IFclk_dec_disable	Div ratio	***PLL_IFclk_div***	PLL input (IFDIV)	Comment
1	xN	N	N	Max is 63 decimal
0	x1	1	61	decimal
	x2	2	62	
	x4	4	63	

x6	6	2
xN	N	N-4
x64	64	60

PLL_TXclk_div (Read/Write) 0xD000_0064

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0001															
description	reserved							TXclk_dec_disable	reserved			Pll_TXclk-div				

Pll_TXclk-div (divider) – (div x1 .. x16)
TXclock divider configuration. Possible divide ratio: 1,2,4,6…64

TXclk_Dec_dis – decoder disable – When high, the Pll_TXclk-div routed directly to PLL.
When low, the Pll ***TX***clk-div gets the decoder output.

TXclk_dec_disable	Div ratio	*PLL_TXclk_div*	PLL input (TXDIV)	Comment
1	xN	N	N	Max is 63 decimal
0	x1	1	61	decimal
	x2	2	62	
	x4	4	63	
	x6	6	2	
	xN	N	N-4	
	X16	64	60	

pll_div_en (Read/Write) 0xD000_0066

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
description	reserved													Pll_TXclk_ en	Pll_IFclk_ en	Pll_sclk_ en

Pll_TXclk_en - TXclock enable signal.
Pll_IFclk_en - IFclock enable signal.
Pll_Sclk_en - Sclock enable signal.

pll_div_bypass (Read/Write) 0xD000_0068

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x000F															
description	reserved												Pll_VCO_ byp	Pll_TXclk_ byp	Pll_IFclk_ byp	Pll_Sclk_ byp

Pll_VCO_byp - VCO bypass signal.
Pll_TXclk_byp – Select the EXTCLK to be routed to PLL TXclk.
Pll_IFclk_byp - Select the EXTCLK to be routed to PLL IFclk.
Pll_Sclk_byp - Select the EXTCLK to be routed to PLL Sclk.

pll_div_bypass_rst (Read/Write) 0xD000_006A

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
description	reserved												Pll_VCO_ Byp_rst	Pll_TXclk_ Byp_rst	Pll_IFclk_ Byp_rst	Pll_Sclk_ Byp_rst

Pll_VCO_byp_rst - Reset to the PLL VCO bypass mux. Verifies that the EXTCLK can go through the mux (self reset bit). When write ‘1’ to this bit, the PLL_VCO_BYP_RST signal goes high for 4 clock cycles.
Note: PLL[vco_byp_rst] = ***Pll_VCO_Byp_rst*** | (!io_rstn)

Pll_TXclk_byp_rst - Reset to the PLL TXclk bypass mux. Verifies that the EXTCLK can go through the mux (self reset bit). When write ‘1’ to this bit, the PLL_TXclk_BYP_RST signal goes high for 4 clock cycles.
Note: PLL[TXclk_byp_rst] = ***Pll_TXclk_Byp_rst*** | (!io_rstn)

Pll_IFclk_byp_rst - Reset to the PLL IFclk bypass mux. Verifies that the EXTCLK can go through the mux (self reset bit). When write ‘1’ to this bit, the PLL_IFclk_BYP_RST signal goes high for 4 clock cycles. Note: PLL[IFclk_byp_rst] = ***Pll_IFclk_Byp_rst*** | (!io_rstn)

pll_Sclk_byp_rst - Reset to the PLL Sclk bypass mux. Verifies that the EXTCLK can go through the mux (self reset bit). When write ‘1’ to this bit, the PLL_Sclk_BYP_RST signal goes high for 4 clock cycles.
Note: PLL[Sclk_byp_rst] = ***Pll_Sclk_Byp_rst*** | (!io_rstn)

pll_div_update (Read/Write) 0xD000_006C

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
description	reserved															pll_div_ Update

pll_div_update - PLL dividers update (self reset bit).
When write ‘1’ to this bit, the PLL goes high for 4 clock cycles.

Core_memory(Read/Write) 0xD000_0F10

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Core_memory[15:0]															

Host may Read/Write 48K Core Rom memory and 12K high memory by access to this address. The access start-address is defined by address 0x0020-0x0022 for Write and by address 0x0024-0x0026 for Read.

Core_memory_cnt(Read/Write) 0xD000_0F12

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0000*															
description	Core_memory[15:0]															

Host may Read/Write 48K Core memory and 12K high memory by access to this address. The access start-address is defined by address 0x0028-0x002a for Write, and by address 0x002c-0x002e for Read.

CONT REGISTERS(0XD000_1000 ~ 0XD000_1FFF)

int0*(Read/Write)* *0xD000_1000*

Bits	7	6	5	4	3	2	1	0
Default	0x1							
description								Int0

int0- host interrupt

Core_status1(Read/Write)* *0xD000_1002

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Core_status1[15:0]															

Core_status2(Read/Write)* *0xD000_1004

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Core_status2[15:0]															

Core status1/2 are Debug registers!!!!!

chip_id_ (READ only)* *0xD000_1006

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x05BA*															
description	**Chip ID – 5BA**															

chip_version (READ only)* *0xD000_1008

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default									*0xA0*							
description									**Chip version – A0**							

CHIP_ID 05BA
CHIP_VERSION A0

Gpio_data_out (Write/Read)* *0xD000_100A

Bits	8	7	6	5	4	3	2	1	0
Default	*0x00*								
description							GPIO3 - DATA	GPIO2 - DATA	GPIO1 - DATA

GPIOx_data – When GPIOx_mode is set to 'GPO' the GPIOx port will contain the GPIOx_data out[x] contains.

Gpio_data_In (Read only)* *0xD000_100C

Bits	8	7	6	5	4	3	2	1	0

Default	*0x00*			
description		GPIO3 - DATA	GPIO2 - DATA	GPIO1 - DATA

GPIOx_data – When GPIOx_mode is set to 'GPI' the GPIOx port will be sampled to Gpio_data_In[x].

Gpio's_mode (Read/Write) 0xD000_100E

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0xFFFF*															
description	**reserved**				**GPIO3 mode**				**GPIO2 mode**				**GPIO1 mode**			

GPIOx mode - 0000 - STRB_OUT.

0001 - **Tri state**.

0010 - **Tri state**.

0011 - **Tri state**.

0100 - **Tri state**.

0101 - **Tri state**.

0110 - **Tri state**.

0111 - Master I2C (**gpio1= mclk, gpio2 = mdat)**.

1000 – GPO mode.

1001 – GPI mode.

1010 **– STRB_IN.**

1011 **– XENON (gpio1 only).**

1111 **– Tri state**

adr_tst_pin_toggeling_capture (Read Only) 0xD000_1010

Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0000*															
description	**reserved**			**Dis_ sw_ conf**	**memsel**				**reserved**			**TST_MODE**				

[4:0] TST_MODE = Caputre the 5 (0-4) first toggling of pin TST after RSTN.

When tst_mode = 0x0 - SW configuration: not perform any test setup (Normal)
When tst_mode = 0x4 - SW configuration: TST_PVI
When tst_mode = 0x07 - SW configuration: TST_PLL
When tst_mode = 0x0d - SW configuration: TST_MIPI_ANA
When tst_mode = 0x0e - SW configuration: TST_IDD

[11:8] memsel = Caputre the 4 (5-8) toggeling of pin TST .

When (tst_mode = 0x4) and (memsel = 0x1) - SW configuration: TST_MIPI_HS1
When (tst_mode = 0x4) and (memsel = 0x2) - SW configuration: TST_MIPI_HS2
When (tst_mode = 0x4) and (memsel = 0x3) - SW configuration: TST_MIPI_LP1
When (tst_mode = 0x4) and (memsel = 0x4) - SW configuration: TST_MIPI_LP2
When (tst_mode = 0x4) and (memsel = 0x5) - SW configuration: TST_STBY

[12] Disable_SW_configuration - When ‘1’ – SW will not setup any test configuration.

adr_jtag_bypass (Read/Write) 0xD000_1012

Bits	7	6	5	4	3	2	1	0
Default	0x00							
description	reserved							adr_jtag_bypass

adr_jtag_bypass – When writing ‘1’ – the jtag interface is bypassed toward the ARM with no synchronization.

Tst_modes_ (Read/Write) 0xD000_1014

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Cnt_tst_modes															

TST_PLL = cnt_tst_modes[1]
TST_PVI = cnt_tst_mode[2]
TST_CID = cnt_tst_modes[3]
TST_CIS = cnt_tst_modes[4]
TST_IDD = cnt_tst_modes[5]
TST_ARM_JTAG = cnt_tst_modes[6]
Reserved cnt_tst_modes[7]
TST_MONITOR = cnt_tst_modes[8]
TST_FPGA = cnt_tst_modes[9]
Reserved cnt_tst_modes[10]
Reserved cnt_tst_modes[11]
TST_LPBK_B = cnt_tst_modes[12]
TST_PLLBYP = cnt_tst_modes[13]
TST_MIPI_ANA = cnt_tst_modes[14]
TST_CIS_11BIT = cnt_tst_modes[15]

Tst_monitor_ (Read/Write) 0xD000_1016

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*															
description	Tst_monitor															

Test port - Enables output debug registers.

Monitor_select (Read/Write) 0xD000_1018

Bits	15 - 1	0
Default	*0x0000*	
description	Reserved	Tst_monitor_sel

Test port - Enables tst_monitor output debug registers.

SW_STRB_IN_ (Read/Write) 0xD000_101C

Bits		0
Default	*0x0000*	
description		**Sw_strb_in**

SW_STRB_IN - Software Strobe In

ROM_HALF_CLK_MODE (Read/Write) 0xD000_1026

Bits		0
Default	*0x0001*	
description		**Rom_half_clk_mode**

rom_half_clk_mode – when set Rom gets clock with half rate of ARM clock. and generate single wait state after each read.

GPIO_Rise_fall_n (Read/Write) 0xD000_1028

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default																*0x1*
description																GPIO_rise_fall_n

GPIO_rise_fall_n – when set GPIO interrupt will detect rising edges of GPI, when low GPIO interrupt will detect falling edges of GPI (0x100C).

Int0 (Read/Write) 0xD000_1030

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								host interrupt

Int0 mask (Read/Write) 0xD000_1032

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								host interrupt_mask

Host_interrupt - The Host interface can activate an interrupt by writing to the Host_interrupt address

Int2 (Read/Write) 0xD000_1038 - GPIO

Bits	8	7	6	5	4	3	2	1	0
Default	*0x0*								
description							**GPIO3**	**GPIO2**	**GPIO1**

Int2 mask (Read/Write) 0xD000_103A

Bits	8	7	6	5	4	3	2	1	0
Default	*0xF*								
description							**GPIO3**	**GPIO2**	**GPIO1**

Int2– The Int2 signal that goes to the core is the OR bitwise on the entire vector. In order to reset a single bit the core should write `1` to this bit.

Int3 (Read/Write) 0xD000_103C - END_COLOR - VOUT

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description							Isp end	Vout end

Int3 mask (Read/Write) 0xD000_103E

Bits	7	6	5	4	3	2	1	0
Default	*0x3*							
Description							Isp end	Vout end

ISP_end_int - The ISP preview sends a signal notifies that end of frame is detected and all frame data has been sent by the ISP.

Vout_end_int - The VOUT sends an interrupt notifies that end of frame is detected and all frame data has been sent out output interface.

Int7 (Read/Write) 0xD000_104C

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description								Timer0

Int7 mask (Read/Write) 0xD000_104E

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								Timer0

Timer0 - Timer0 interrupt reflect that the Timer counter has finished the counting.

Int7 – the Int7 signal that goes to the core is the OR bitwise on the entire vector. In order to reset a single bit the core should write `1` to this bit.

Int8 (Read/Write) 0xD000_1050

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description								Timer1

Int8 mask (Read/Write) 0xD000_1052

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								Timer1

Timer0 - Timer0 interrupt reflect that the Timer counter has finished the counting.

Int8 – the Int8 signal that goes to the core is the OR bitwise on the entire vector. In order to reset a single bit the core should write `1` to this bit.

Int9 (Read/Write) 0xD000_1054 - FLICKER

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description								Flicker_req

Int9_mask (Read/Write) 0xD000_1056

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								Flicker_req

Flicker_req - Flicker interrupt for every line statistics.

Int9 – the Int9 signal that goes to the core is the OR bitwise on the entire vector. In order to reset a single bit the core should write `1` to this bit.

Int10 (Read/Write) 0xD000_1058 - EVEN THUMBNAL STATISTIC

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description								thstat even

Int10 mask (Read/Write) 0xD000_105A

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								thstat even

Thstate even strip finished

Int11 (Read/Write) 0xD000_105C - THUMBNALI STATISTIC

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description								thstat odd

Int11_mask (Read/Write) 0xD000_105E

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								thstat odd

Thstate odd strip finished

Int12 (Read/Write) 0xD000_1060 - TG vector[13:0]

Bits	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*													
description	TG	TG	TG	TG	TG	TG	TG	TG	TG	TG	TG	TG	TG	TG

	vec [13]	vec [12]	vec [11]	vec [10]	vec [9]	vec [8]	vec [7]	vec [6]	vec [5]	vec [4]	vec [3]	vec [2]	vec [1]	vec [0]

Int12_mask (Read/Write) 0xD000_1062

Bits	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x3FFF*													
description	TG vec [13]	TG vec [12]	TG vec [11]	TG vec [10]	TG vec [9]	TG vec [8]	TG vec [7]	TG vec [6]	TG vec [5]	TG vec [4]	TG vec [3]	TG vec [2]	TG vec [1]	TG vec [0]

Interrupt Name	Bit	Description
TG frame start	TG_vector[0]	Indicates Vertical and Horizontal counters zero.
TG frame end	TG_vector[1]	Indicates Vertical and Horizontal counters last value.
TG line start	TG_vector[2]	Indicates Horizontal counters zero.
TG line int0	TG_vector[3]	Indicates Vertical counter equals interrupt register_0.(immediate)
TG line int1	TG_vector[4]	Indicates Vertical counter equals interrupt register_1.(non immediate)
TG active start	TG_vector[5]	Indicates start of active visible lines readout.
TG active end	TG_vector[6]	Indicates end of active visible lines readout.
TG LE Rout start	TG_vector[7]	Indicates start of long exposure readout.
TG SE Rout start	TG_vector[8]	Indicates start of short exposure readout.
TG Global_array_reset	TG_vector[9]	Indicates the timing of global array reset. (All lines were globally reset)
TG Vsync start	TG_vector[10]	Indicates Vsync signal start.
TG all line integrated	TG_vector[11]	Indicates integration event of last image line.
TG last long readout	TG_vector[12]	Indicates finish of long readout.
TG last short readout	TG_vector[13]	Indicates finish of short readout.

Int15 (Read/Write) 0xD000_106C - TG strobe ({out/in}{fe/re})

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	**0x00**							
Description					TG strobe in RE	TG strobe in FE	TG strobe out RE	TG strobe out FE

Int15 mask (Read/Write) 0xD000_106E

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	**0x0F**							
Description					TG strobe in RE	TG strobe in FE	TG strobe out RE	TG strobe out FE

TG strobe out/in – Falling Edge/Rising Edge

PLL – RB Register (Not interrupt) 0xD000_1070

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Default	*0x0*							
Description							PLL_lock_fall	PLL_lock_rise

PLL mask (Read/Write) 0xD000_1072

Bits	7	6	5	4	3	2	1	0
Default	*0x3*							
Description							PLL_lock_fall	PLL_lock_rise

PLL_lock_fall **-** Event Pll lock falling edge
PLL_lock_rise **–** Event Pll lock rising edge

MIPI – RB Register (Not interrupt) 0xD000_1074

Bits	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x0000*														
description															**Outif_errors**

Test port - Enables output debug registers.

MIPI_mask (Read/Write) 0xD000_1076

Bits	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x7FFF*														
description															**Outif_errors**

[14] - ileaver_error_video_underflow
[13] - ileaver_error_jpeg_underflow
[12] - ileaver_error_emb_underflow
[11] - ileaver_error_missed_event
[10] - ileaver_error_events_overflow
[9] - ileaver_error_syncbuf_overflow
[8] - pack_video_error_overflow
[7] - pack_video_error_rw_contention
[6] - pack_video_error_size
[5] - pack_jpeg_error_overflow
[4] - pack_jpeg_error_rw_contention
[3] - pack_emb_error_overflow
[2] - pack_emb_error_rw_contention
[1] - csi2llp_error_starvation_sync
[0] - csi2llp_error_inconsistence_sync

PVI – RB Register (Not interrupt) 0xD000_1078

Bits	8	7	6	5	4	3	2	1	0
Default	**0x00**								
Description							**pvi_lpckt_err_sync**	**pvi_spckt_err_sync**	**pvi_seq_err_sync**

PVI mask (Read/Write) 0xD000_107A

Bits	8	7	6	5	4	3	2	1	0
Default	0x7								
Description							**pvi_lpckt_err_sync**	**pvi_spckt_err_sync**	**pvi_seq_err_sync**

MI2C – RB Register (Not interrupt) 0xD000_107C

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description							mi2c_transfer_end	mi2c_fifo_int

MI2C mask (Read/Write) 0xD000_107E

Bits	7	6	5	4	3	2	1	0
Default	*0x3*							
Description							mi2c_transfer_end	mi2c_fifo_int

mi2c_transfer_end - The end transfer interrupt is sent when the master I2C completes sending or receiving the data according to its programming.

mi2c_fifo_int - The FIFO interrupt is set when MI2C FIFO is half occupancy

D0_D4_cs12 [11:0] (Read/Write) 0xD000_1082

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x155*											
Description			D4_cd10		D3_cd10		D2_cd10		D1_cd10		D0_cd10	

***D0_cd10* [1:0] – D0 pad cd1/0 control {*D0_cd1, D0_cd0*}.**

00 – 2mA

01 – 4mA

10 – 6mA

11 – 8mA

***D1_cd10* [1:0] – D1 pad cd1/0 control {*D1_cd1, D1_cd0*}.**

***D2_cd10* [1:0] – D2 pad cd1/0 control {*D2_cd1, D2_cd0*}.**

***D3_cd10* [1:0] – D3 pad cd1/0 control {*D3_cd1, D3_cd0*}.**

***D4_cd10* [1:0] – D4 pad cd1/0 control {*D4_cd1, D4_cd0*}.**

D9_D5_cd12 [11:0] (Read/Write) 0xD000_1084

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x155*											
Description			D9_cd10		D8_cd10		D7_cd10		D6_cd10		D5_cd10	

***D5_cd10* [1:0] – D5 pad cd1/0 control {*D5_cd1, D5_cd0*}.**

***D6_cd10* [1:0] – D6 pad cd1/0 control {*D6_cd1, D6_cd0*}.**

***D7_cd10* [1:0] – D7 pad cd1/0 control {*D7_cd1, D7_cd0*}.**

***D8_cd10* [1:0] – D8 pad cd1/0 control {*D8_cd1, D8_cd0*}.**

***D9_cd10* [1:0] – D9 pad cd1/0 control {*D9_cd1, D9_cd0*}.**

GPIO_cd10 [11:0] (Read/Write) 0xD000_1086

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x55*											
Description							GPIO3_cd10		GPIO2_cd10		GPIO1_cd10	

***GPIO1_cd10* [1:0] – *GPIO1* pad cd1/cd0 control { *GPIO1_cd1, GPIO1_cd0*}.**

***GPIO2_cd10* [3:2] – *GPIO2* pad cd1/cd0 control { *GPIO2_cd1, GPIO2_cd0*}.**

***GPIO3_cd10* [5:4] – *GPIO3* pad cd1/cd0 control { *GPIO3_cd1, GPIO3_cd0*}.**

CLKs_cd10 [11:0] (Read/Write) 0xD000_1088

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x555*											
Description		Sda_cd10		Scl_cd10		PCLK_cd10		Reserved		Vsync_cd10		Hsync_cd10

Hsync _cd10* [1:0] – *Hsync* pad cs1/0 control { *Hsync _cd1, Hsync _cd0}.

Vsync _cd10* [1:0] – *Vsync* pad cs1/0 control { *Vsync _cd1, Vsync _cd0}.

PCLK _cd10* [1:0] – *PCLK* pad cs1/0 control { *PCLK _cd1, PCLK _cd0}.

Scl _cd10* [1:0] – *Scl* pad cs1/0 control { *Scl _cd1, Scl _cd0}.

Sdat _cd10* [1:0] – *Sdat* pad cs1/0 control { *Sdat _cd1, Sdat _cd0}.

Cnt_normal_output_en [11:0] (Read/Write) 0xD000_108A

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x1111*												
Description													

d0_en = cnt_normal_output_en[0];

d1_en = cnt_normal_output_en[1];

d2_en = cnt_normal_output_en[2];

d3_en = cnt_normal_output_en[3];

d4_en = cnt_normal_output_en[4];

d5_en = cnt_normal_output_en[5];

d6_en = cnt_normal_output_en[6];

d7_en = cnt_normal_output_en[7];

d8_en = cnt_normal_output_en[8];

d9_en = cnt_normal_output_en[9];

hsync_en = cnt_normal_output_en[10];

vsync_en = cnt_normal_output_en[11];

pclk_en = cnt_normal_output_en[12];

Cnt_normal_output_ien [11:0] (Read/Write) 0xD000_108C

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x1111*												
Description													

d0_en = cnt_normal_output_ien[0];

d1_en = cnt_normal_output_ien[1];

d2_en = cnt_normal_output_ien[2];

d3_en = cnt_normal_output_ien[3];

d4_en = cnt_normal_output_ien[4];

d5_en = cnt_normal_output_ien[5];

d6_en = cnt_normal_output_ien[6];

d7_en = cnt_normal_output_ien[7];

d8_en = cnt_normal_output_ien[8];

d9_en = cnt_normal_output_ien[9];

hsync_en = cnt_normal_output_ien[10];

vsync_en = cnt_normal_output_ien[11];

pclk_en = cnt_normal_output_ien[12];

sel_gtg_types(Read/Write) 0xD000_1090

Bits	8	7	6	5	4	3	2	1	0
Default	*0x155*								
Description	Sel_gtg_types_bypass	Sel_gtg_line_type				Sel_gtg_col_type			

Sel_gtg_col_type - Select the gtg-col type to be capture. Default =4’h5 [5BA], which is active pixels code.

Sel_gtg_line_type - Select the gtg-line type to be capture. Default =4’h5 [5BA], which is active lines code.

Sel_gtg_types_bypass – bypass all pixels to isp chain. Default 1’b1 – no filtering.

sel_bayer_tg_isp_n (Read/Write) 0xD000_1092

Bits	7	6	5	4	3	2	1	0
Default	*0x1*							
Description								sel_bayer_tg_isp_n

When ‘1’ – The Bayer for OUTIF is coming from TG output (after sensor_fe blocks).

When ‘1’ - The Bayer for OUTIF is coming from GRAS output bus.

Pclk_delay_r (Read/Write) 0xD000_1094

Bits	7	6	5	4	3	2	1	0
Default	*0x0*							
Description	Pclk_delay_r							

chip_id_ (READ only) 0xD000_1096

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x05BA*															
description	**Chip ID – 5BA**															

chip_version (READ only) 0xD000_1098

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	*0x00*															
description	**Chip version – 00**															

CHIP_ID 05BA
CHIP_VERSION 00

TP_enable0 [15:0] (Read/Write) 0xD000_1100

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	**0x0000**															
Description	**TP15**	**TP14**	**TP13**	**TP12**	**TP11**	**TP10**	**TP9**	**TP8**	**TP7**	**TP6**	**TP5**	**TP4**	**TP3**	**TP2**	**TP1**	**TP0**

TP_enable0[15:0] – if bit is set the corresponding Trap comparator is enabled.

Trap_add [15:0] (Read/Write) 0xD000_1110 ~ 0xD000_112E

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
Description	Trap_add															

Trap_add – 0-31 addresses registers related to enable0.

This address can be compared to the address arrives from the core, and if equals it will be replaced with the Patch address.

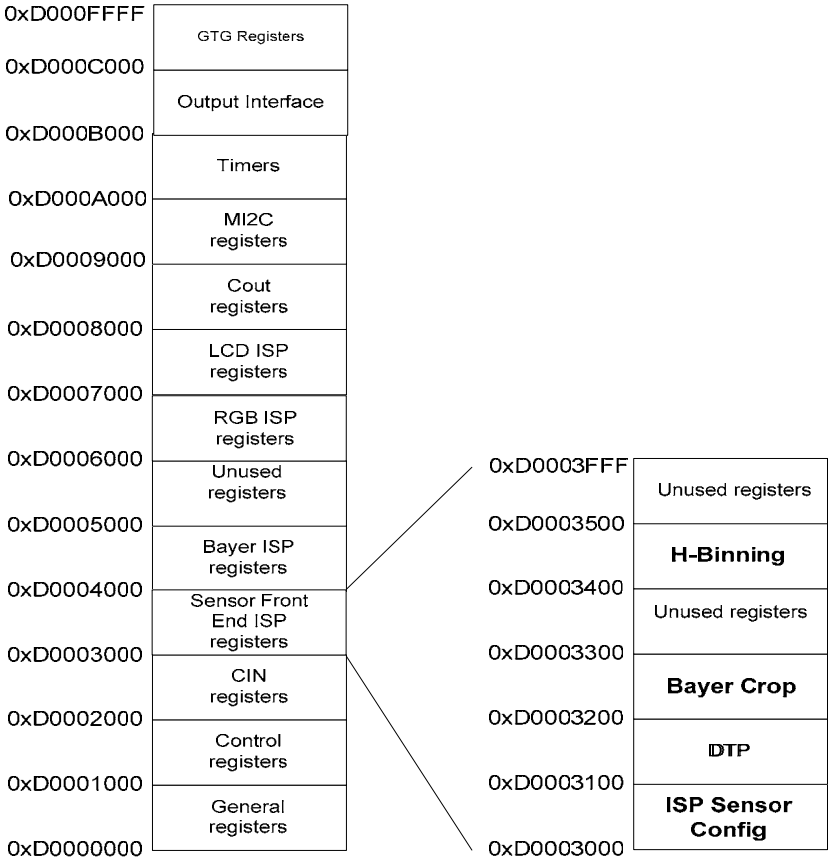
Patch_add [15:0] (Read/Write) 0xD000_1150 ~ 0xD000_116E

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0x0000															
Description	Patch_add															

Patch_add – 0-31 addresses registers related to enable0.

This address can be placed instead of the core address, if the Trap address equals to the core address.

SENSOR FRONT END REGISTERS(0XD000_3000 ~ 0XD000_3FFF)



Disable_ISP_input_data (Read/Write) 0xD000_3000

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
description	Reserved							Disable_ISP_input_data

Disable_ISP_input_data – Bayer Data and Controls zero muxing
0 – Regular data input to GISP
1 – Zero input data/controls to GISP

Bayer Crop

bcrop_bypass 0xD000_3200

Address: *0x00*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	1
Description	reserved							bcrop_bypass

bcrop_start_x 0xD000_3202

Address: *0x02*

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
Description	bcrop_start_x											

crop_start_x - defines the X-coordinate of the button left corner of the output window in the input image

bcrop_start_y 0xD000_3204

Address: *0x04*

Bits	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0
Description	bcrop_start_y										

crop_start_x - defines the Y-coordinate of the button left corner of the output window in the input image

bcrop_size_x 0xD000_3206

Address: *0x06*

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	1	0	1	0	0	0	0	0	0	0	0
Description	bcrop_size_x											

bcrop_size_x - defines the image size X-coordinate of the output window

bcrop_size_y 0xD000_3208

Address: *0x08*

Bits	10	9	8	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0	0	0	0
Description	bcrop_size_y										

bcrop_size_y - defines the iamge size Y-coordinate of the output window

DTP (Digital Test Pattern Generator)

dtp_mode (Read/Write) 0xD000_3100

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
description	Reserved					Dtp_mode		

Dtp_mode - Controls the output of the test pattern module.
0 – no pattern
1 – solid color
2 – 100% color bar
3 – fade to grey color bars
4 – PN9
5 – PN12
6 – Macbeth

dtp_solid_red (Read/Write) 0xD000_3102

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_solid_red											

Dtp_solid_red - The test data used to replace red pixel data

dtp_solid_green_r (Read/Write) 0xD000_3104

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_solid_green_r											

Dtp_solid_green_r - The test data used to replace green pixel data (on Red line)

dtp_solid_green_b (Read/Write) 0xD000_3106

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_solid_green_b											

Dtp_solid_green_b - The test data used to replace green pixel data (on Blue line)

dtp_solid_blue (Read/Write) 0xD000_3108

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_solid_blue											

Dtp_solid_blue - The test data used to replace blue pixel data

Dtp_cursor_h_width (Read/Write) 0xD000_310A

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_cursor_h_width											

Dtp_cursor_h_width - Defines the width of the horizontal cursor (in pixels)

Dtp_cursor_h_position (Read/Write) 0xD000_310C

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_cursor_h_position											

Dtp_cursor_h_position - Defines the top edge of the horizontal cursor (actual value is value+1 since first pixel is zero)

Dtp_cursor_v_width (Read/Write) 0xD000_310E

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_cursor_v_width											

Dtp_cursor_v_width - Defines the width of the vertical cursor (in pixels)

Dtp_cursor_v_position (Read/Write) 0xD000_3110

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	dtp_cursor_v_position											

Dtp_cursor_v_position - Defines the left-hand edge of the vertical cursor.
A value of 0xFFFF switches the vertical cursor into automatic mode where it automatically advances every frame. (actual value is value+1 since first line is zero)

Dtp_bit_mask (Read/Write) 0xD000_3112

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
description	Bit_Mask_pattern	Bit_mask											

Defines ancillary configurations for test pattern modes.
bit_mask – mask for 12-bit bayer.
Bit_mask_pattern – bit mask pattern.
0 - masked bits replaced with 0.
1 - masked bits replaced with 1.

Dtp_color_mask (Read/Write) 0xD000_3114

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
description	Reserved			Color_Mask_pattern	color_mask			

Defines ancillary configurations for test pattern modes.
color_mask – color masking (3=GreenR, 2=Red, 1=GreenB, 0=Blue)(1=synthetic image, replace with fix value, 0=remain original value).
color_mask_pattern – color mask pattern.
0 - masked colors replaced with 0.
1 - masked colors replaced with 1.

Dtp_fade_aux (Read/Write) 0xD000_3116

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
description	reserved					Fade_aux		

Defines ancillary configurations for test pattern modes.
Fade_aux– number of LSBs for quanization (Zero padding only)

Dtp_size_x (Read/Write) 0xD000_3118

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	Dtp_Size_x											

Dtp_Size_x – Defines X-size of out-coming synthetic image.
.

Dtp_size_y (Read/Write) 0xD000_311A

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0
description	Dtp_Size_y											

Dtp_Size_y – Defines Y-size of out-coming synthetic image.

Notes:

1. Skip-to feature allowed.
2. No special bypass bit indicating if processed or non-processed test-pattern will be send out via PVI. Binning enable/disable is enough to play the role.
5. The colors of Macbeth are standard only globally. Slight deviations can appear from one Macbeth color chart to another.
6. Windowing allowed. Dedicated local registers (sizeX, sizeY) define synthetic picture size.

FirstPixelFirstLineColor (Read/Write) 0xD000_311C

Bits	10	9	8	7	6	5	4	3	2	1	0
Default										0	0
Description	Bit[0] - Controls the color of first line in frame (0 – red, 1 - blue). Bit[1] - Controls the color of first pixel in first line (0 – green, 1 – red/blue).										

HBinning

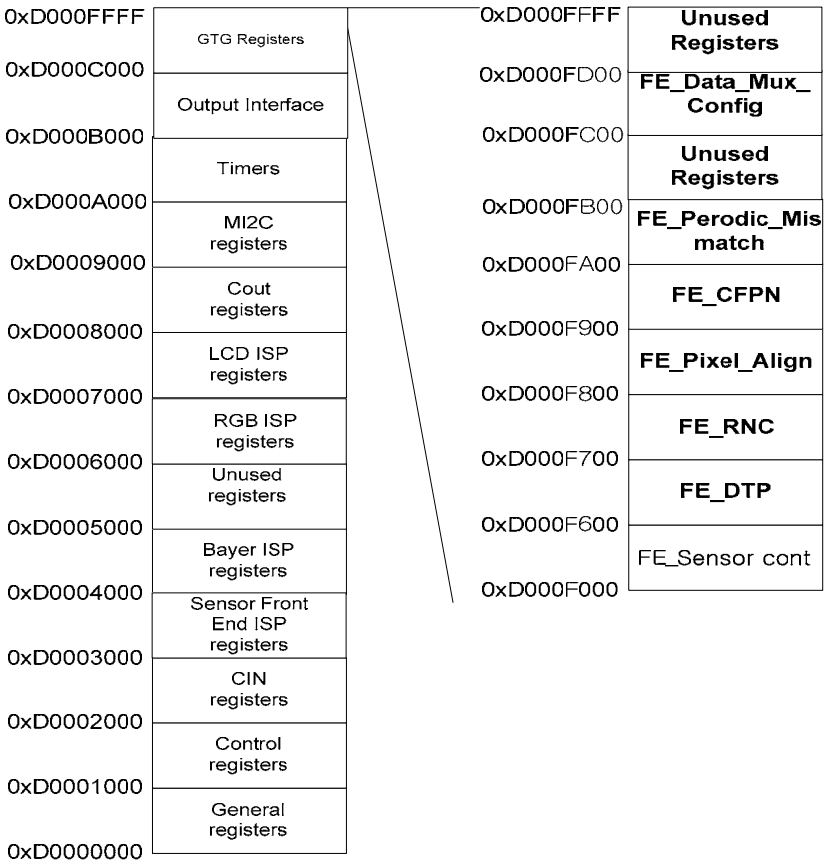
hbin_control 0xD000_3400

Address: *0x00*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	1	0	0
Description	reserved					binning_divide2	binning_active	binning_bypass

Asserted binning_active -> average of two horizontal adjacent pixels from the same color. Asserted binning_divide2 -> division by two on the binning result
Note that binning_bypass=0 and binning_active=0 is equivalent to binning_bypass=1

SENSOR FRONT END GTG REGISTERS(0XD000_F000 ~ 0XD000_FFFF)



Sensor ISP FE Config

Adc resolution Control 0xD000_FC00

Address: 0x00

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	1	0
description	Reserved					adc_mode	adc_res_sel	

adc_res_sel – Sel ADC resolution width.Bayer Data and Controls zero muxing
0 – 8 Bit data width.
1 – 9 Bit data width.
2 - 10 Bit data width. (default)
3 – N/A
adc_mode – Choose least or most significant bists (only in 9 or 10 bits resolutions).
0 - least significant bists
1 - most significant bists

FEDTP

fedtp_config 0xD000_F600

Address: *0x00*

Bits	7	6	5	4	3	2	1	0

Default	0	0	0	0	0	0	0	0
Description	reserved			fedtp_mode				fedtp_bypass

0 - path through
1 - solid color
2 - gradient
3 - address dependent noise
4 - random
5 - gradient + address dependent noise
6 - gradient + random
7 - out pixel attributes

fedtp_solid_r 0xD000_F602

Address: *0x02*

Bits	9	8	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0	0	0
Description	fedtp_solid_r									

Red value for solid color output

fedtp_solid_gr 0xD000_F604

Address: *0x04*

Bits	9	8	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0	0	0
Description	fedtp_solid_gr									

Green (Gr) value for solid color output

fedtp_solid_gb 0xD000_F606

Address: *0x06*

Bits	9	8	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0	0	0
Description	fedtp_solid_gb									

Green (Gb) value for solid color output

fedtp_solid_b 0xD000_F608

Address: *0x08*

Bits	9	8	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0	0	0
Description	fedtp_solid_b									

Blue value for solid color output

fedtp_gradient 0xD000_F60A

Address: *0x0a*

Bits	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	1	1
Description	fedtp_gradient_			fedtp_gradient_			fedtp_gradient_	fedtp_gradient_	fedtp_gradient_	fedtp_gradient_

	vert_shift	horz_shift	vert_inv	horz_inv	vert	horz

Apply horizontal gradient
Apply vertical gradient
Invert address X
Invert address Y
Scaling factor for X dependent gradient
Scaling factor for Y dependent gradient

fedtp_noise 0xD000_F60C

Address: *0x0C*

Bits	9	8	7	6	5	4	3	2
Default	0	0	0	0	0	0	0	0
Description	fedtp_mode56_shift_right		fedtp_v_noise_shift_right		fedtp_h_noise_shift_right		fedtp_vert_noise	fedtp_horz_n

Right shift the random values
Enable horizontal address dependent noise 6bit
Enable vertical address dependent noise 6bit
Right shift for noise generated values
Right shift for noise generated values
Right shift for modes 5 and 6 generated values

bpradlc

bpradlc_control (Read/Write) 0xD000_F700

Address: *0x00*

Bits	7	6	5	4	3	2	1	0
Default	0	1	0	0	0	0	1	0
Description	bpradlc_nactive_pedestal[2:0]			reserved			bpradlc_passthrough	bpradlc_bypass

bpradlc_nactive_pedestal :
0:disable,
1:32,
2:64,
3:128,
4:256,
else - N.A.

bpradlc_passthrough
Only active and CFPN data passthrough bpradlc block. (not included OB data)

bpradlc_bypass
All input data included OB data bypass bpradlc block. (Only for debugging)

bpradlc_ncfpn_pedestal (Read/Write) 0xD000_F702

Address: *0x02*

Bits	8	7	6	5	4	3	2	1	0
Default	0	0	1	0	0	0	0	0	0
Description	Bpradlc_ncfpnpedestal[8:0]								

Bpradlc_ncfpnpedestal - determine the cfpn data pedestal
The register contain its actual value
0:0,
1:1,
....
64:64,(default)
....
256:256 (max value)

bpradlc_rate_control_enable (Read/Write) 0xD000_F704

Address: *0x04*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	1
Description	reserved							bpradlc_rate_control_enable

bpradlc_rate_control_enable - enables pixel output rate control

bpradlc_pixels_gap (Read/Write) 0xD000_F706

Address: *0x06*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	1
Description	reserved	bpradlc_pixels_gap						

bpradlc_pixels_gap - defines pixel output rate at the end of lines and for two last lines.
Relevant while bpradlc_rate_control_enable bit [0x0e] is set. Otherwise,
pixels will be outputed every clock cycle

Bpradlc_f_adlc_tune_r(Read/Write) 0xD000_F708

Address: *0x08*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_f_adlc_tune_r[7:0]							

Frame OB (R channel) data offset control register

bpradlc_f_adlc_tune_gr (Read/Write) 0xD000_F70A

Address: *0x0a*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_f_adlc_tune_gr[7:0]							

Frame OB (Gr channel) data offset control register

bpradlc_f_adlc_tune_gb (Read/Write) 0xD000_F70C

Address: *0x0c*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_f_adlc_tune_gb[7:0]							

Frame OB (Gb channel) data offset control register

bpradlc_f_adlc_tune_b (Read/Write) 0xD000_F70E

Address: *0x0e*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_f_adlc_tune_b[7:0]							

Frame OB (B channel) data offset control register

bpradlc_f_adlc_tune_total (Read/Write) 0xD000_F710

Address: *0x10*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_f_adlc_tune_total[7:0]							

All Frame OB data offset control register

Bpradlc_line_adlc_tune_r(Read/Write) 0xD000_F712

Address: *0x12*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_line_adlc_tune_r[7:0]							

Line OB (R channel) data offset control register

bpradlc_line_adlc_tune_gr (Read/Write) 0xD000_F714

Address: *0x14*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_line_adlc_tune_gr[7:0]							

Line OB (Gr channel) data offset control register

bpradlc_line_adlc_tune_gb (Read/Write) 0xD000_F716

Address: *0x16*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_line_adlc_tune_gb[7:0]							

Line OB (Gb channel) data offset control register

bpradlc_line_adlc_tune_b (Read/Write) 0xD000_F718

Address: *0x18*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_line_adlc_tune_b[7:0]							

Line OB (B channel) data offset control register

bpradlc_line_adlc_tune_total (Read/Write) 0xD000_F71A

Address: *0x1A*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_line_adlc_tune_total[7:0]							

All Line OB data offset control register

bpradlc_adlc_en (Read/Write) 0xD000_F71C

Address: *0x1C*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	reserved					bpradlc_f_l_adlc_en	bpradlc_f_adlc_en	bpradlc_h_adlc_en

bpradlc_f_l_adlc_en – Line ADLC enable register for Frame OB data 0b:disable, 1b:enable

bpradlc_f_adlc_en – Frame ADLC enable register 0b:disable, 1b:enable

bpradlc_h_adlc_en – Line ADLC enable register 0b: disable, 1b: enable

bpradlc_adlc_option (Read/Write) 0xD000_F71E

Address: *0x1E*

Bits	7	6	5	4	3	2	1	0

Default	0	0	0	0	0	0	0	0
Description	reserved			bpradlc_h_adlc_ch_sel	bpradlc_max_data_clip_sel[1:0]		bpradlc_f_adlc_max_data_clip_sel[1:0]	

bpradlc_h_adlc_ch_sel – Line ADLC channel seperation option selection register

0 : not seperating, 1 : seperating

bpradlc_max_data_clip_sel[1:0] – Line ADLC maximum value limitation register

00 : Line ADLC data < 128

01 : Line ADLC data < 256

10 : Line ADLC data < 512

bpradlc_f_adlc_max_data_clip_sel[1:0] – Frame ADLC maximum value limitation register

00 : Frame ADLC data < 128

01 : Frame ADLC data < 256

10 : Frame ADLC data < 512

bpradlc_adlc_bpr_en (Read/Write) 0xD000_F720

Address: *0x20*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	reserved							bpradlc_adlc_bpr_en

bpradlc_adlc_bpr_en – All ADLC data BPR(1-D Bad pixel replacement) enable register

0: disable, 1: enable

bpradlc_adlc_bpr_thresh (Read/Write) 0xD000_F722

Address: *0x22*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	bpradlc_adlc_bpr_thresh[7:0]							

bpradlc_adlc_bpr_thresh[7:0] – BPR threshould value for detecting bad pixels

When (D(n-1) – D(n)) value and (D(n+1)-D(n) value are lager than bpradlc_adlc_bpr_thresh[7:0] register value, then internal logic operates BPR algorithm.

CFPN

cfpn_config 0xD000_F900

Address: *0x00*

Bits	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	1	0	0	0	1	0	1
Description	cfpn_ref_gain			cfpn_ref_lines		cfpn_ hbinning	cfpn_output_direct	cfpn_overflow_protect	cfpn_ passthrough	cfpn_ bypass

0x1 - Bypassing correction, all incoming data bypassed to output,
0x0 - No bypassing, performing CFPN correction (unless passthrough mode is set)
0x1 - incoming data bypassed to output, but MinCFPN, MaxCFPN, MeanRef and CFPN data are calculated
divide incoming data by 2 to prevent overflow during MeanRef calculation
Output direct lines during CFPN collect data - can be enabled in regular operation or passthrough
cfpn X binning - 0 - no binning
1 - binning 1/2
Number of reference lines to collect for MeanRef calculation
Number of reference lines = 2^(5+ref_lines)
0x0 - 32
0x1 - 64
0x2 - 128
0x3 - 256
Reference Gain during direct lines = 2^(ref_gain)
0x0 - 1
0x1 - 2
0x2 - 4
0x3 - 8
0x4 - 16
0x5 - 32
0x6 - 64
0x7 - reserved

cfpn_mean 0xD000_F902

Address: *0x02*

Bits	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	1	0	0	0	0	0	0	0	0	0	0
Description	cfpn_num_mean_cols		cfpn_mean_start									

Pixel address to start mean columns calculation
Number of columns to perform calculation on them
Number of columns = 2^(7+num_mean_columns)
0x0 - 128
0x1 - 256
0x2 - 512
0x3 - 1024

cfpn_act_start_column 0xD000_F904

Address: *0x04*

Bits	7	6	5	4	3	2	1	0
Default	1	0	1	0	0	1	1	0
Description	cfpn_act_start_column							

Start address of active columns

cfpn_crnt_gain 0xD000_F906

Address: *0x06*

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	1	0	0	0	0	0
Description	cfpn_crnt_gain												

Current analog gain 8.5 (8-integer, 5-fraction)

cfpn_min_max (Read only) 0xD000_F90A

Address: *0x0A*

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	Max_FPN								Min_FPN							

Max_FPN - Maximal detected FPN (signed value – 2's complement)
Min_FPN - Minimal detected FPN (signed value – 2's complement)

cfpn_rate_control_enable (WRITE only) 0xD000_F90C

Address: *0x0C*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	reserved							cfpn_rate_control_enable

cfpn_rate_control_enable - enables pixel output rate control

cfpn_pixels_gap (WRITE only) 0xD000_F90E

Address: *0x0E*

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	reserved	cfpn_pixels_gap						

cfpn_pixels_gap - defines pixel output rate at the end of lines and for two last lines.
Relevant while cfpn_rate_control_enable bit [0x0e] is set.
Otherwise, pixels will be outputed every clock cycle.

MSM (Periodic Offset)

msm_modes (Read/Write) 0xD000_FA00

Bits	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0
Description	reserved				gain_shift			msm_bypass

shift left of: [up_down_limit(bayer +offset)]*gain

msm_offset00 (Read/Write) 0xD000_ FA 02

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset00												

offset for bayer input. 2's complement

msm_offset01 (Read/Write) 0xD000_ FA04

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset01												

offset for bayer input. 2's complement

msm_offset10 (Read/Write) 0xD000_ FA06

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset10												

offset for bayer input. 2's complement

msm_offset11 (Read/Write) 0xD000_ FA08

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset11												

offset for bayer input. 2's complement

msm_offset20 (Read/Write) 0xD000_FA0A

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset20												

offset for bayer input. 2's complement

msm_offset21 (Read/Write) 0xD000_FA0C

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset21												

offset for bayer input. 2's complement

msm_offset30 (Read/Write) 0xD000_FA0E

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset30												

offset for bayer input. 2's complement

msm_offset31 (Read/Write) 0xD000_FA10

Bits	12	11	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0
Description	msm_offset31												

offset for bayer input. 2's complement

msm_limit 0xD000_FA12

Address: *0x12*

Bits	10	9	8	7	6	5	4	3	2	1	0
Default	0	0	0	0	0	0	0	0	0	0	0
Description	msm_limit										

IMPORTANT NOTICE

The information furnished by Samsung Electronics in this document is believed to be accurate and reliable. However, no responsibility is assumed by Samsung Electronics for its use, nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Samsung Electronics. Samsung Electronics reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current and complete.

For More Information
E-mail: jtjoo@samsung.com
http://www.samsung.com/Products/Semiconductor/index.htm