UM3507

User manual

Configuring the 5.1 MP rolling and global shutter monochrome image sensor

Introduction

This document is the user manual for the VB5943 and VD5943 5.1 MP image sensors. It should be read in conjunction with the VB5943 and VD5943 datasheet (DS14968) and reference manual (RM0532). Note that the reference manual provides a complete list of register addresses. The aim of this document is to provide further technical details about the VB5943 and VD5943 sensors, and to provide guidance on how to configure them for different applications.

1 Acronyms and abbreviations

Table 1. Acronyms and abbreviations

Acronym/abbreviation	Definition
CCITT	consultative committee for international telephony and telegraphy
CRC	cyclic redundancy check
CSI	camera serial interface
DI	data identifier
DT	data type
ECC	error correcting code
EOF	end of frame
EWS	electrical wafer sort
FPS	frame per second
FSM	finite state machine
FW	firmware
FWP	firmware patch
GPH	group parameter hold (register)
GS	global shutter
HDR	high dynamic range
ISL	intelligent status line
ISP	image signal processing
MCU	microcontroller unit
NVM	nonvolatile memory
OIF	output interface
PCB	printed circuit board
PLL	phase-locked loop
PWL	piece-wise linear
PWM	pulse-width modulation
RS	rolling shutter
SCL	serial clock line
SDA	serial data line
SDR	standard dynamic range
SS	subsampling
UI	user interface
VC	virtual channel
VT	video timing

2 Sensor interface

Information on the application schematic and the layout recommendations are provided in the product datasheet. This section explains the integration of the image sensors into a system from a hardware perspective.

2.1 Power supplies

To power on the device, all the external supplies (VCORE, VDDIO, VANA) must be properly provided according to the device characteristics described in the section *Electrical characteristics* of the datasheet. Each supply should be in the operating range for a standard application configuration and compatible with the device power consumption (average and peak current).

2.2 Hardware reset (RESETn)

The device chip enable is managed by the mode of the RESETn pin:

- At low level, the device is in HW_STANDBY state (power down).
- At high level, the device is active if all the power supplies are present.

Refer to the datasheet for the electrical specifications of the RESETn pin.

2.3 Input clock (EXTCLK)

The devices need an input clock to operate correctly. Refer to section *Electrical characteristics* of the datasheets for the constraints of the input clock.

2.4 I^{2}C interface

2.4.1 CCI protocol

The devices are controlled via an I^{2}C interface as defined by the MIPI CSI-2 specification (refer to *MIPI Alliance standard for camera serial interface* 2). The devices use CCI protocol over I^{2}C with 2 bytes subaddresses.

The value of the pull-up resistors, on the SDA and SCL signals, must be chosen to ensure that the I^{2}C bus constraints are met in fast mode+. If the device is used in fast mode, the correct pull-up resistors must be selected to support fast mode and fast mode+. The default configuration is fast mode+.

2.4.2 Data alignment within registers

Registers larger than 8 bits wide are stored LSB (least significant byte) first. That is, the LSB is placed in the location with the lowest index (little endian).

Note: *This ordering of multi-byte registers is contrary to the recommended byte ordering detailed in the MIPI CSI-2 specification, which states that multi-byte registers should be stored MSB first.*

2.4.3 I^{2}C device address change

The default device address, 0x20 (including R/W bit) can be overridden when the sensor is in SW_STANDBY state. The new I^{2}C address must be stored in the 8-bit field DEVICE_ID. Once set, the application of dedicated command I2_COMMS_UPDATE changes the I^{2}C address of the sensor. This address modification lasts until the sensor is reset or a new command for changing the I^{2}C address is applied.

Table 2. DEVICE_COMMS_CTRL [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DRIVE			DEVICE_ID							

2.4.4 I²C acknowledge strength

The field DRIVE configures the drive strength of the I²C transistors for acknowledge events. The default configuration is fast+ mode

Table 3. DRIVE field values

DRIVE I2C mode	Current drive	I2C mode
0	4 mA	Fast
6	20 mA	Fast+

2.4.5 Known limitation and workaround

The sensor's I²C interface does not support single or sequential reads that start from the current location. Consequently, you must use a repeated start condition to read the registers.

Figure 1. Incompatible I²C read sequence

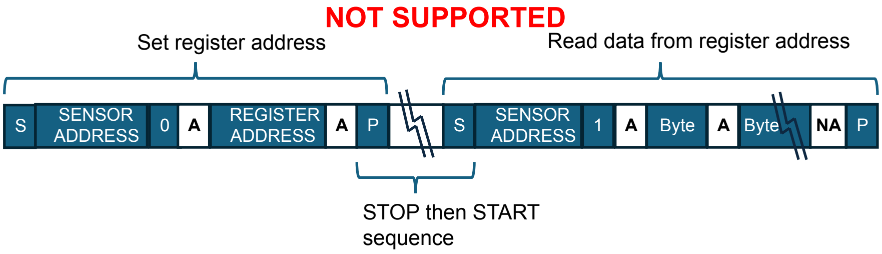
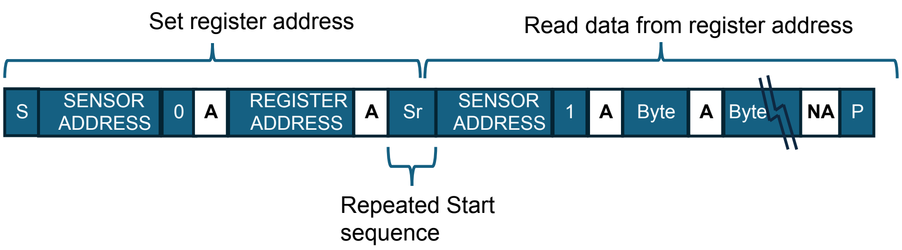


Figure 2. Reliable I²C read sequence



2.5 MIPI CSI-2 transmitter interface

The output interface of the device outputs is compliant with MIPI CSI-2 v1.1. The following characteristics are supported:

- Scalable MIPI lane number (quad lane, dual lane, or single lane set up).
- Data and clock polarity can be inverted.
- Lanes can be physically remapped to simplify the PCB layout.

This configuration is detailed in Section 12: Output interface.

2.6 GPIO interface

The VB5943 and VD5943 sensors have four GPIOs, which synchronize signals or act as in/out general purpose pins.

3 Control interface

The sensor presents to the host a user interface (bank of registers), accessible via the I²C bus.

The host writes high-level parameters in the user interface, and an MCU inside the sensor applies the proper sensor configuration based on UI settings provided by the host.

The firmware managing this MCU is located in a ROM but may have to be updated before streaming (using a firmware patch). This consists of a sequence of data transmissions over the I²C bus.

3.1 Register default values

The register default value depends on the device state. The default value, after device reset, may differ from the register default values observed once the firmware is in SW STANDBY state.

Unless noted otherwise, the default values listed in this document are the default values observed in SW STANDBY state after the boot command is complete.

3.2 Register groups

The registers are grouped together based on their functional purpose.

Figure 3. Register groups overview

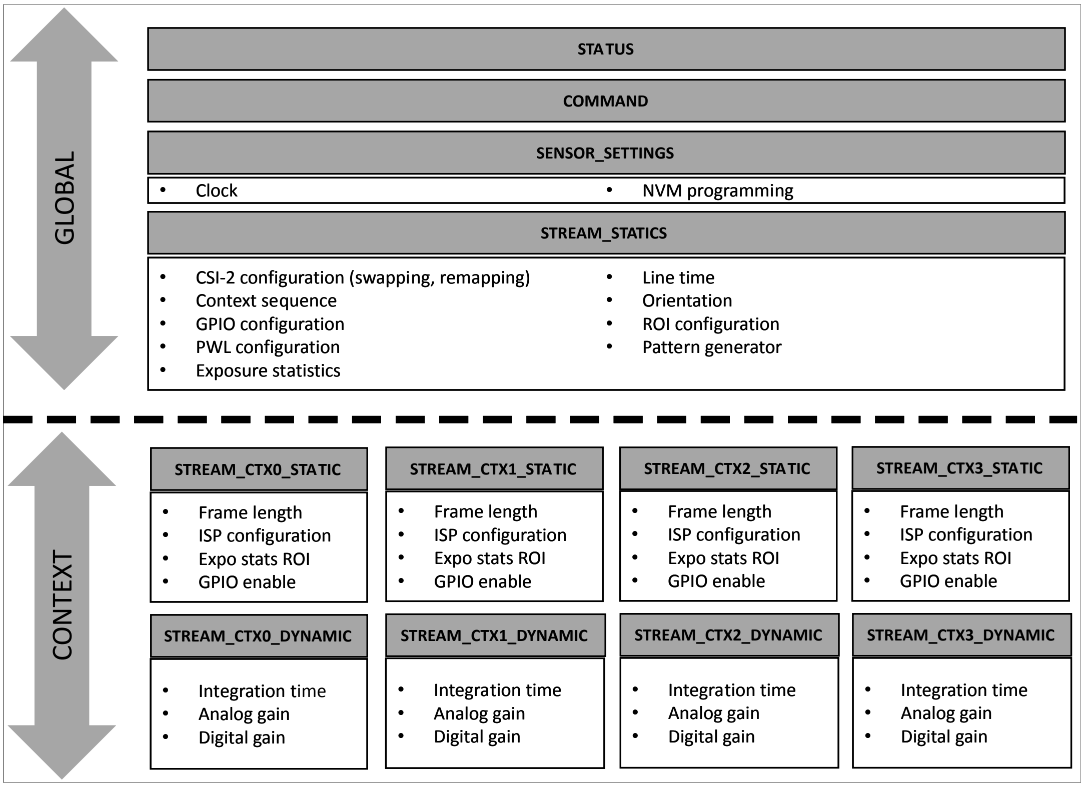


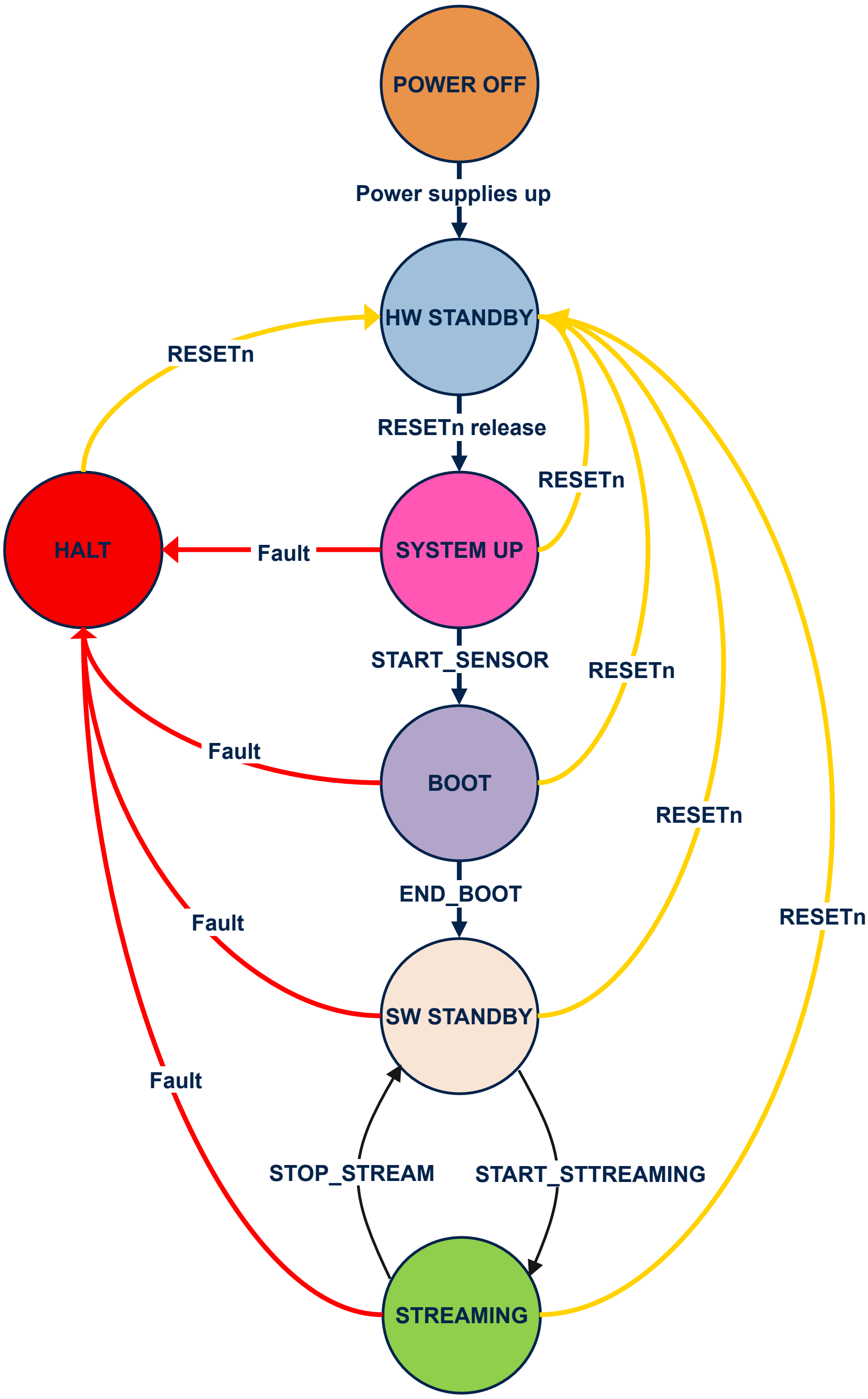
Table 4. Register group description

Group	Base address	Access	Description
STATUS	0x0000	RO	General status of the device
COMMAND	0x0514	R/W	Commands to control the device
SENSOR SETTINGS	0x0734	R/W	General settings of the device
STREAM_STATICS	0x090C	R/W	Global static configuration for the streaming
STREAM_CXT0_STATIC	0x0B40	R/W	Used to control the static settings for each dedicated context. These settings cannot be updated during streaming
STREAM_CXT1_STATIC	0x0B88	R/W	
STREAM_CXT2_STATIC	0x0BD0	R/W	
STREAM_CXT3_STATIC	0x0C18	R/W	
STREAM_CXT0_DYNAMIC	0x0C78	R/W	Used to control the dynamic settings for each dedicated context. These settings can be updated during streaming
STREAM_CXT1_DYNAMIC	0x0CA0	R/W	
STREAM_CXT2_DYNAMIC	0x0CC8	R/W	
STREAM_CXT3_DYNAMIC	0x0CF0	R/W	
NVM MIRROR	0x0EC4	R/W	Buffer for NVM operations
FW PATCH	0x2000	R/W	Buffer to load the firmware patch

4 Firmware state machine

4.1 Description

Figure 4. Firmware state machine



4.2 Status

The *SYSTEM_FSM* register in the STATUS group indicates the global state of the device (FSM).

Table 5. **SYSTEM_FSM_STATE [7:0]**

7	6	5	4	3	2	1	0
VALUE							

Table 6. **List of states for SYSTEM_FSM**

Value	Name	Description
0	HW_STBY	HW_STANDBY - the RESETn signal is asserted.
1	SYSTEM_UP	SYSTEM UP - this is the boot preparation.
2	BOOT	BOOT
3	SW_STBY	SW_STANDBY - the boot has been completed. The sensor is ready to stream.
4	STREAMING	STREAMING - the video stream is enabled.
6	HALT	HALT - an error has occurred. A reset is mandatory.

5 Commands

Commands are sent by the host to four specific registers of the sensor UI (COMMAND group). Each register is enabled during a specific state of the FSM. Once a command is executed, the register is cleared to acknowledge the command. Any new command is ignored until the current command is acknowledged.

Table 7. SYSTEM_UP/BOOT/SW_STBY/STREAMING [7:0]

7	6	5	4	3	2	1	0
COMMAND							

5.1 SYSTEM UP command register

Table 8. SYSTEM UP command description

Command	Name	Description
0	CMD_ACK	Sensor is idle
1	START_SENSOR	Initiates the boot

5.2 BOOT command register

Table 9. BOOT command descriptions

Command	Name	Description
0	CMD_ACK	Sensor is idle
2	LOAD_FWP	Starts processing the FW patch
16	END_BOOT	Finalizes the boot procedure

5.3 SW STANDBY command register

Table 10. SW STANDBY command descriptions

Command	Name	Description
0x0	CMD_ACK	Sensor is idle
0x1	START_STREAMING	Starts the video stream
0x2	THSENS_READ	Forces an update of the status reported by the thermal sensor
0x3	UPDATE_VT_RAM_START	Loads an update for the video timing
0x4	UPDATE_VT_RAM_END	Enables the updated video timing
0x15	NVM_READ	Reads the NVM (data are available in the NVM mirror)
0x16	NVM_PROG	Programs the NVM (from the NVM mirror)
0x18	I2C_COMMS_UPDATE	Assign to sensor a new non-permanent I^2C address

5.4 STREAMING command register

Table 11. **STREAMING command descriptions**

Command	Name	Description
0	CMD_ACK	The sensor is idle
1	STOP_STREAM	Stops the video stream after completion of the ongoing frame.

6 Boot sequence

This section describes the sequence from power-up to SW STANDBY state.

6.1 Device power up sequence

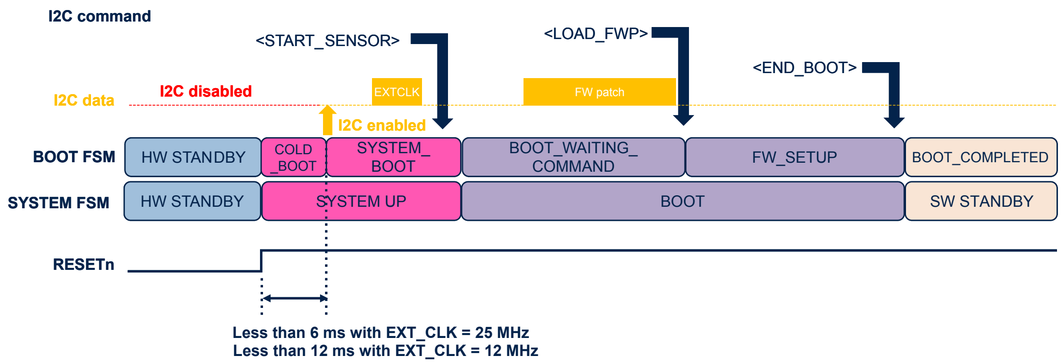
To power on the devices, all the external supplies (VCORE, VDDIO, VANA) must be properly provided as described in the section *Electrical characteristics* of the datasheet. RESETn must stay low. The devices are in HW_STBY state.

6.2 Boot diagram

Considering an I^2C frequency above 400 kHz, the required duration from releasing RESETn pin to SW_STANDBY state can be lower than:

- 12 ms with an external clock at 25 MHz
- 18 ms with an external clock at 12 MHz

Figure 5. Boot diagram



Note: The LOAD_FWP command is required only if an MCU firmware patch is loaded.

7 NVM

The NVM is a one-time programmable memory that allows customers to store their own information inside the sensor. The customer area is composed of 75 words, including the I^2C device address.

The NVM includes two areas that contain two types of information:

- EWS area: Information about the product (example: traceability and calibration)
- Customer area: Information about the customer

NVM access is granted through the UI using the dedicated commands NVM_READ and NVM_WRITE. These commands are available with the register in SW STANDBY state.

The NVM is organized in 32-bit words. Each word is independently protected by an ECC mechanism. It is possible to write either one word or several words (burst) during a single operation. However, it is not possible to partially write a word, or to write the same word several times.

7.1 NVM register descriptions

7.1.1 Sensor settings

Three registers from the SENSOR_SETTINGS register group configure the NVM operations.

Table 12. **NVM_CTRL [7:0]**

7	6	5	4	3	2	1	0
0						1	0

Note that the NVM_CTRL must be set at **0x02**.

Table 13. **NVM_START_ADDR [7:0]**

7	6	5	4	3	2	1	0
VALUE							

VALUE is the index of the first word to read/write. It must be within the range 0 to 74. Index 0 is used to change the device I^2C address.

Table 14. **NVM_NB_OF_WORDS [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							VALUE								

Value is a 9-bit number to read/write. It must be within the range 1 to 75.

7.1.2 NVM custom mirror

The NVM is not directly accessible by the host. The data are transferred through the NVM_CTM_MIRROR register group. Following a write command (or a read command), the FW transfers data from the UI to the NVM (or respectively from the NVM to the UI).

Table 15. NVM_CTM_MIRROR [31:0]

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Index 0 - I^{2}C device address</td></tr>
<tr><td colspan="32">Index 1 – Customer word 1</td></tr>
<tr><td colspan="32">Index 2 – Customer word 2</td></tr>
<tr><td colspan="32">. . .</td></tr>
<tr><td colspan="32">Index 73 – Customer word 73</td></tr>
<tr><td colspan="32">Index 74 – Customer word 74</td></tr>
</table>

7.1.3 NVM status

Table 16. NVM status register

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">Reserved</td><td>ECC_CORRECTION</td><td colspan="8">NB_ECC</td><td colspan="8">ERROR_ADDRESS</td><td colspan="4">ERROR</td><td colspan="4">OPERATION</td></tr>
</table>

Table 17. NVM bitfield description

Field	Value	Description
OPERATION	0x0 = NOT_USED	
	0x5 = OPERATION_SUCCESS	NVM operation has succeeded
	0xF = OPERATION_FAIL	NVM operation has failed
ERROR	0x0 = NO_ERROR	
	0x1 = BAD_CONFIG	
	0x2 = READ_FAIL	
	0x3 = PROG_FAIL	
	0x4 = ECC_CORRECTION_FAIL	
	0x5 = PROG_ON_NON_VIRGIN_WORD	
ERROR_ADDRESS		Index of the field following the field in error. Faulty address is ERROR_ADDRESS - 1
NB_ECC		Number of errors detected
ECC_CORRECTION	0 = NO_CORRECTION	No error
	1 = DETECTED	An error has been detected and corrected

7.2 NVM programming

The sequence to program the memory is described below:

- Write the data in the NVM_CTM_MIRROR register group. The index in the NVM_CTM_MIRROR corresponds to the index in the NVM.
- Set the NVM_CTRL to 0x02, and set the NVM_CTRL, NVM_START_ADDR, and NVM_NB_OF_WORDS.
- Execute the NVM_WRITE command (SW_STBY command register).
- The operation completes when the SW_STBY command register is reset (CMD_ACK).
- Check the NVM status register: OPERATION = OPERATION_SUCCESS.

7.3 NVM reading

The memory is automatically read at boot and transferred to the NVM_CTM_MIRROR register group. If the NVM_CTM_MIRROR has been modified, it is possible to read the NVM again. The sequence to read the memory is described below:

- Set the NVM_CTRL to 0x02, and set the NVM_CTRL, NVM_START_ADDR, and NVM_NB_OF_WORDS.
- Execute the NVM_READ command (SW_STBY command register).
- The operation completes when the SW_STBY command register is reset (CMD_ACK).
- Check the NVM status register.
- Check the NVM status register: OPERATION = OPERATION_SUCCESS.
- Read the data in the NVM_CTM_MIRROR register group.

7.4 I²C device address update

The default sensor address is 0x20 (bit [7:0]). It is possible to change the address by programming the NVM. This programming definitively overrides the default value.

The first word of the customer area stores the new address. The same value must be repeated in three subfields of the word. An additional subfield (I2C_KEY = 0xAA) must also be set.

Note: Only the 7-MSB of the new address is provided. Bit 0 (R/W) is discarded.

Table 18. NVM_MIRROR I2C_ADDRESS field

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">I2C_KEY = 0xAA</td><td>0</td><td colspan="7">ADDRESS [7:1]</td><td>0</td><td colspan="7">ADDRESS [7:1]</td><td>0</td><td colspan="7">ADDRESS [7:1]</td></tr>
</table>

The new value becomes active **after** a reset. This new value is reported in a register of the STATUS register group called **I2C_ADDRESS**.

The I2C_ADDRESS provides the 7-bit address, without R/W bit.

Table 19. STATUS I2C_ADDRESS field

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="7">DEVICE_ID</td></tr>
</table>

For example, if register NVM_MIRROR I2C_ADDRESS is programmed with 0xAA12_1212 (I2C address 0x24 including R/W bit), after reset , the status register STATUS I2C_ADDRESS reports 0x12.

8 Timings

8.1 Clocks

The VB5943 and VD5943 devices support a range of external clock source frequencies from 12 MHz to 50 MHz. A PLL is used, and the firmware configures the clock tree according to the external clock value. All internal operating clocks (example: pixel clock) are automatically generated to be as close to the optimal value as possible.

8.2 External clock configuration

The CLK_RANGE pins must be programmed according to the table below at power-up.

Table 20. CLK_RANGE pins

EXTCLK	CLK_RANGE0	CLK_RANGE1	Default EXT_CLK after boot
25 MHz	0	0	25 MHz
12–24 MHz	1	0	24 MHz
24–36 MHz	0	1	36 MHz
36–50 MHz	1	1	50 MHz

Once the automatic built-in self-test has been executed, the I²C communication is enabled. During the boot, if the external clock is not generated at 25 MHz (default value), the exact frequency must be programmed by the host in the EXT_CLOCK register in hertz (SENSOR_SETTINGS group).

Table 21. EXT_CLK [31:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXT_CLK																															

The firmware automatically sets the system PLL and the clock tree according to the external clock value.

8.3 System and pixel clocks

To achieve a target of 1500 MHz, the system PLL is set by the firmware as follows:

$$NDIV = \begin{Bmatrix} FLOOR\left(\frac{1500}{EXTCLK}\right),\ EXTCLK\ < 27MHz \\ FLOOR\left(\frac{3000}{EXTCLK}\right),\ EXTCLK\ \geq\ 27MHz \end{Bmatrix}$$

Then, the PLL output has the following value: $SYSTEM\_PLL\_CLK = \begin{Bmatrix} NDIV\ \times EXTCLK,\ EXTCLK < 27\ MHz \\ NDIV\ /2\ \times\ EXTCLK,\ EXTCLK\ \geq 27\ MHz \end{Bmatrix}$

Eventually, the pixel clock has the following value:

$$PIXEL\_CLK = \frac{SYSTEM\_PLL\_CLK}{16}$$

The devices are designed to run with a pixel clock at 93.75 MHz (see Table 23. Clocks and PLL configuration examples). The clock values are reported in the STATUS register group (see Table 22. Clock values in the STATUS register group).

Table 22. Clock values in the STATUS register group

Register name	Bits	Values
SYSTEM_PLL_CLK	[31:0]	PLL output frequency in Hz
PIXEL_CLK	[31:0]	PIXEL clock frequency in Hz

Table 23. Clocks and PLL configuration examples

EXTCLK	SYSTEM_PLL_CLK	PIXEL_CLK
12 MHz	1500 MHz	93.75 MHz
24 MHz	1488 MHz	93.00 MHz
25 MHz	**1500 MHz**	**93.75 MHz**
27 MHz	1498.5 MHz	93.66 MHz
50 MHz	1500 MHz	93.75 MHz

8.4 MIPI clock

A dedicated PLL is in charge of generating the clock supplied to the MIPI interface.

The minimum data rate is 600 Mbps per lane. However, using the configuration combining rolling shutter and HDR, the minimum data rate is limited to 8 times the pixel clock. For example, 750 Mbps per lane when PIXEL_CLK is running to its default 93.75 MHz value).

The real MIPI data rate is proportional to EXT_CLK when EXT_CLK is less than 27 MHz, or EXT_CLK/2 when equal or greater than 27 MHz.

When EXT_CLK is less than 27 MHz:

REAL_MIPI_DATARATE_PER_LANE = integer_part_of (MIPI_DATA_RATE / EXT_CLK) *EXT_CLK

When EXT_CLK is equal or greater than 27 MHz:

REAL_MIPI_DATARATE_PER_LANE = integer_part_of (MIPI_DATA_RATE / (EXT_CLK/2))*EXT_CLK/2

Refer to Section 12.1.1: Output data rate for a description of the field MIPI_DATA_RATE.

Multiplying the value of status register CLK_MIPI_PLL_MULT by EXT_CLK is giving the real MIPI data rate (or EXT_CLK/2 when EXT_CLK is equal or greater than 27 MHz).

Table 24. CLK_MIPI_PLL_MULT [7:0] status register

7	6	5	4	3	2	1	0
CLK_MIPI_PLL_MULT							

To avoid an unexpected lowered MIPI data rate after firmware computation, the requested MIPI data rate shall be slightly increased by the host (to cross the EXT_CLK step truncation threshold).

Note: *For EXT_CLK equal to or greater than 27 MHz and MIPI_DATA_RATE over 750 Mbps, the 8-bit internal register 0x93d8 (MIPI PLL divider) must be programmed by host prior to the START_STREAMING command.*

The value to write in this byte register is MIPI_DATA_RATE / (EXT_CLK/2).

As an example with EXT_CLK = 28 MHz and a target MIPI_DATA_RATE of 1000 Mbps, the value to write is 71 (1000/(28/2)=71).

8.5 Line time

Line time is a global parameter (common to all contexts) controlled by the LINE_LENGTH register in the STREAM_STATICS group. Integration times and frame length are multiples of the line time.

Table 25. LINE_LENGTH [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE_LENGTH															

LINE_LENGTH must be a multiple of 4.

$$line\ time(\mu s) = \frac{LINE\_LENGTH}{PIXEL\_CLK(MHz) \times 4}$$

The default LINE_LENGTH value is 3372. Assuming a default pixel clock of 93.75 MHz, the corresponding default line time is 8.992 µs (see table below).

Table 26. Line time examples

EXTCLK	PIXEL_CLK	LINE_LENGTH	Line time
12 MHz	93.75 MHz	3372	8.992 µs
24 MHz	93.00 MHz	3372	9.065 µs
25 MHz	**93.75 MHz**	**3372**	**8.992 µs**
27 MHz	93.66 MHz	3372	9.001 µs
50 MHz	93.75 MHz	3372	8.992 µs

The minimum line duration is 8.992 µs (see **bold text** in table above).

The maximum recommended line time duration should not exceed **13.5 µs** (see table below). Increasing the line length above this value may reduce pixel performance.

Table 27. Maximum recommended line time examples

EXTCLK	PIXEL_CLK	LINE_LENGTH	Line time
12 MHz	93.75 MHz	5060	13.493 µs
24 MHz	93.00 MHz	5020	13.495 µs
25 MHz	**93.75 MHz**	**5060**	**13.493 µs**
27 MHz	93.66 MHz	5056	13.496 µs
50 MHz	93.75 MHz	5060	13.493 µs

Example for defining the minimum allowed line length

IMAGE_WIDTH = 2560 (columns)
IMAGE_HEIGHT = 1984 (lines)
FIFO_LINE_NB_MAX = 400 (lines)
LINE_TIME_MIN = 9.0 (µs)
PIX_CLK_MHZ = 93.75 (MHz)
PIXEL_RESOLUTION = 10 (bits)
MIPI_LANE_NB = 4
INTER_PACKET_DELAY = 20
REAL_MIPI_DATARATE_PER_LANE_MBPS = 1200 (Mbps)
MIPI_LP_HS_LP_US = 0.25 (µs)

$$LINE\_LENGTH = ((IMAGE\_WIDTH \times PIXEL\_RESOLUTION \div MIPI\_LANE\_NB + 8 \times INTER\_PACKET\_DELAY) \div REAL\_MIPI\_DATARATE\_PER\_LANE\_MBPS + MIPI\_LP\_HS\_LP\_US) \times PIX\_CLK\_MHZ$$

If in global shutter mode:

$$LINE\_LENGTH = 2 \times LINE\_LENGTH \times IMAGE\_HEIGHT \div (IMAGE\_HEIGHT + FIFO\_LINE\_NB\_MAX)$$

If in rolling shutter and HDR mode:

$$LINE\_LENGTH = LINE\_LENGTH + 10$$

If the LINE_LENGTH ÷ PIX_CLK_MHZ is < LINE_TIME_MIN:

$$LINE\_LENGTH = LINE\_TIME\_MINxPIX\_CLK\_MHZ$$

Program 4 x LINE_LENGTH in the register LINE_LENGTH [15:0].

8.6 Frame period

The frame period is set for each of the four contexts. The frame period is controlled by the FRAME_LENGTH register in the STREAM_CTXx_STATIC group. This value is an integer expressed in number of lines. If the line time is modified, the frame period is modified accordingly.

Frame period = FRAME_LENGTH x line time

Table 28. FRAME_LENGTH [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRAME_LENGTH															

Based on the different clock and timing settings, the status register FRAME_RATE provides the frame rate computed by the CPU (8 bit integer value).

If frame rate is above 255 frames per second, a roll over occurs in the status field.

Table 29. FRAME_RATE[15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FRAME_RATE							

8.7 Synchronization mode

The synchronization mode of the device is controlled by the VT_CTRL register from the STATICS_STREAM group. The default mode is LEADER: All operations are automatically scheduled and optimized. This mode is initiated by the command START_STREAMING.

The frame rate is defined by the FRAME_LENGTH register.

The second mode is FOLLOWER where each start of integration is triggered by a pulse on a GPIO.

FOLLOWER mode with hardware triggering

SYNC_MODE must be set to HW_FOLLOWER

One and only one of the four GPIOs must be configured in FSYNC_IN mode (refer to Section 13: GPIOs).

Before the first triggering pulse, the user must send the I^2C command START_STREAMING.

A new integration starts when a pulse is generated on the selected GPIO in FSYNC_IN mode.

The maximum frame rate is defined by the FRAME_LENGTH register.

Table 30. VT_CTRL [7:0]

7	6	5	4	3	2	1	0
Reserved						SYNC_MODE	

Table 31. SYNC mode descriptions

SYNC_MODE	Name	Description
0	LEADER	The device sends video frames continuously
1	HW_FOLLOWER	Each start of integration is triggered by a pulse on the FSYNC_IN.

9 Context management

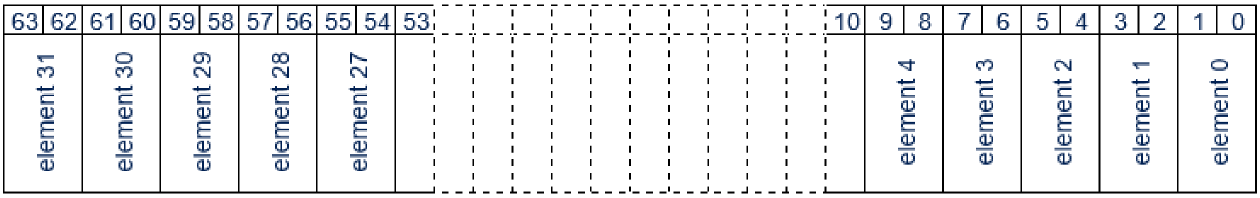
9.1 Sequence vector

The VB5943 and VD5943 sensors have four contexts. Each stores a specific sensor configuration for each sensor. The context configuration registers are split into two subsets:

- The *STREAM_CTXx_STATIC* group contains all the parameters that are updated in SW STANDBY state.
- The *STREAM_CTXx_DYNAMIC* group contains all the parameters that are updated either in SW STANDBY state or in STREAMING state.

During a video stream session, any of the four contexts can be selected to generate a frame. The context sequence is programmed by the CONTEXT_SWITCH_SEQUENCE_VECTOR register. This register is a global parameter and belongs to the STREAM_STATICS group. The register has 32 elements of 2 bits.

Figure 6. CONTEXT_SWITCH_SEQUENCE_VECTOR [63:0]



The value of each element (0, 1, 2 or 3) is related to the context number (0, 1, 2 or 3).

Table 32. Element value

Element value	Context used by the FW to configure the sensors
0	Context 0
1	Context 1
2	Context 2
3	Context 3

The firmware configures the sensor to generate the first frame according to the context stored by the first element (element 0), the second frame according to the second element, and so on. After the last element (element 31), the firmware returns to the first element and generates the next frame according to the first element, then the second element and so on. The streaming is continuous until the reception of the STOP_STREAM command.

To execute a shorter loop, it is possible to set the CONTEXT_SWITCH_LOOP_ELEMENT register. This register is a global parameter and belongs to the STREAM_STATICS group.

Once the firmware has reached and executed the element index corresponding to the CONTEXT_SWITCH_LOOP_ELEMENT value, the firmware returns to the first element to generate the next frame. The range of this register is 0 to 31, to respectively loop on the first element or last element of the sequence vector.

Table 33. CONTEXT_SWITCH_LOOP_ELEMENT [7:0]

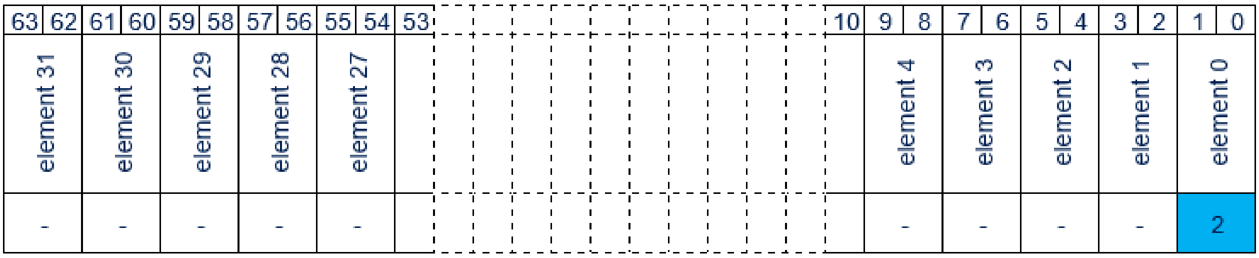
7	6	5	4	3	2	1	0
Reserved			VALUE				

9.2 Examples

9.2.1 Single context

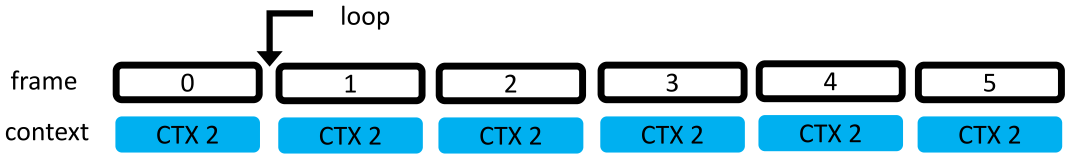
In this use case, only the context 2 is executed. The sequence is made up of one single element (CONTEXT_SWITCH_LOOP_ELEMENT = 0).

Figure 7. **CONTEXT_SWITCH_SEQUENCE_VECTOR**



Description of the sequence at sensor output:

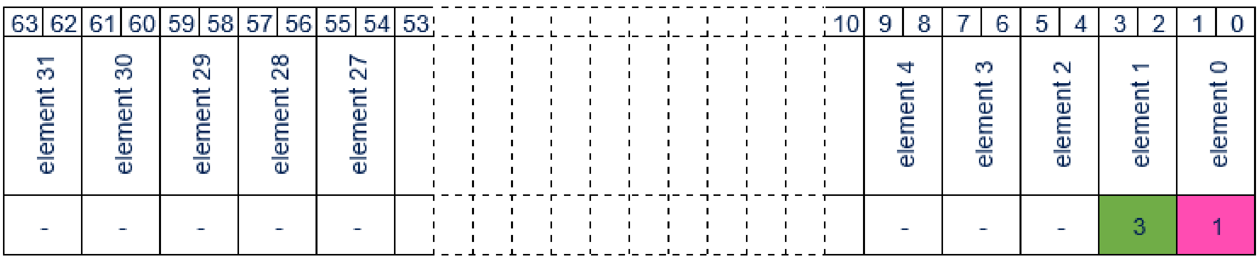
Figure 8. **Single context**



9.2.2 Double context - balanced

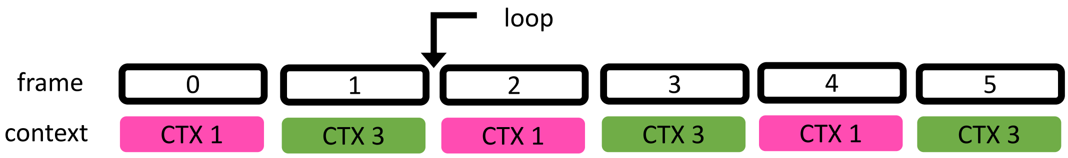
In this use case, context 1 and context 3 are sequentially executed. The sequence is made up of two elements (CONTEXT_SWITCH_LOOP_ELEMENT = 1).

Figure 9. **CONTEXT_SWITCH_SEQUENCE_VECTOR**



The figure below is a description of the sequence at sensor output.

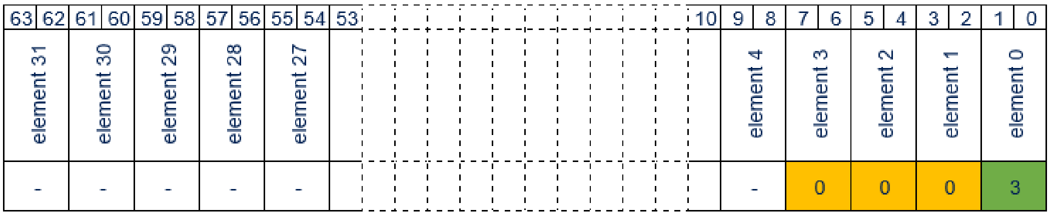
Figure 10. **Double context balanced**



9.2.3 Double context - unbalanced

In this use case, an unbalanced sequence of context 0 and context 3 is executed, but the occurrence of context 0 is three times higher than context 3. The complete sequence has four elements (CONTEXT_SWITCH_LOOP_ELEMENT = 3).

Figure 11. **CONTEXT_SWITCH_SEQUENCE_VECTOR**



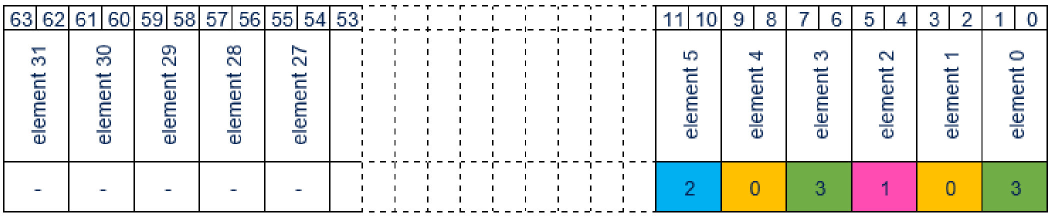
The figure below is a description of the sequence at sensor output.

Figure 12. **Double context unbalanced**

9.2.4 Four contexts - unbalanced

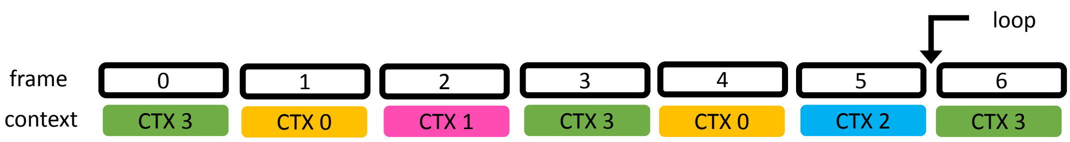
In this use case, an unbalanced sequence of six frames is achieved (CONTEXT_SWITCH_LOOP_ELEMENT = 5). The sequence is made of four contexts.

Figure 13. **CONTEXT_SWITCH_SEQUENCE_VECTOR**



The figure below is a description of the sequence at sensor output.

Figure 14. **Four contexts**



9.3 Context sequence and FPS

The computation of the fps is typical for single context use cases. It is based on the frame length (in seconds):

$$fps = \frac{1}{frame\ length}$$

However, with a multicontext configuration, the calculation of the fps of each context is more complex. Each frame duration has an impact on the fps of the other contexts. This means that increasing a single frame length decreases the fps of all contexts in the sequence.

The following steps are needed to compute the fps of each context:

- Compute the **sequence duration** using the equation below. It adds the frame length of all active elements of the context sequence (from element 0 to *CONTEXT_SWITCH_LOOP_ELEMENT*).
  $$sequence\ duration\ = \Sigma_{i\ =\ 0}^{CONTEXT\_SWITCH\_LOOP\_ELEMENT} framelength\,(element\ i\,)$$
- Compute the **sequence fps** using the equation below and the sequence duration calculated above.
  $$sequence\ fps = \frac{1}{sequence\ duration}$$
- Compute the **fps of each context** using the equation below, the sequence fps, and the context occurrence in the sequence.
  *context fps = sequence fps * context occurrence*

Note: The highest fps ratio between two contexts is 31. The first context has 1 element and the second context has 31 elements.

9.4 Finite streaming sequence

When NB_FRAMES_TO_SEND is set to a value different from zero, using the START_STREAMING command, the sensor launches streaming for the given number of frames defined by NB_FRAMES_TO_SEND before going back to SW_STANDBY. This feature is available in leader mode and follower mode.

Table 34. NB_FRAMES_TO_SEND [7:0]

7	6	5	4	3	2	1	0
NB_FRAMES_TO_SEND							

10 Image processing

10.1 Image processing chain

Each context has a register to globally configure the video timing and the processing chain of the sensor. This register belongs to the STREAM_CTXx_STATIC group. This register controls both video timing (pixel matrix controller) and the image processing chain.

Video timing

- Global shutter/rolling shutter
- Subsampling
- Split exposure
- SDR/HDR

Image processing

- PWL

Table 35. **SENSOR_CONFIGURATION [7:0]**

7	6	5	4	3	2	1	0
VALUE							

The register is described in Table 36. Global shutter configuration table and Table 37. Rolling shutter configuration table.

10.1.1 Global shutter configuration

Figure 15. **Image processing for global shutter configuration**

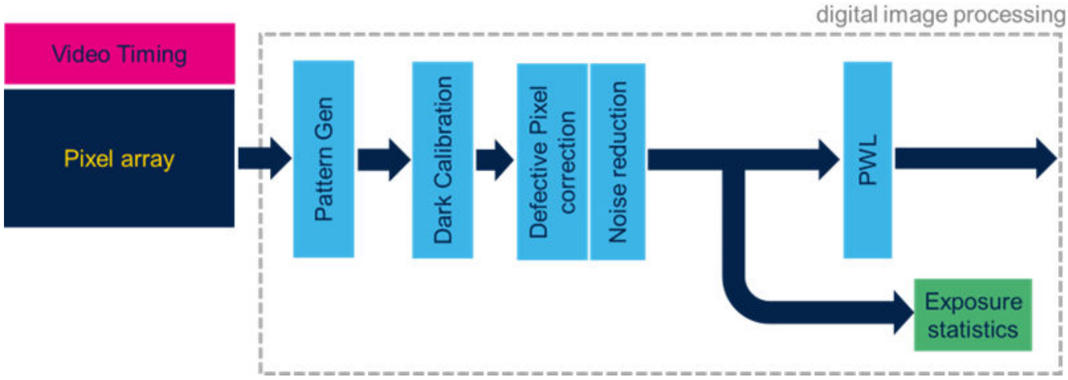


Table 36. **Global shutter configuration table**

Sensor configuration	Video timing configuration			Image processing configuration		Image output	
	VT mode	SS factor	Exposure	Data type format	PWL	Pixel format	Pixel dynamic
0x01	Normal	1	Single	0x2A	10 → 8	P1P2P3P4	8 bits
0x02		1	Single	0x2B	—	P1P2P3P4	10 bits
0x03		1	Split	0x2A	10 → 8	P1P2P3P4	8 bits
0x04		1	Split	0x2B	—	P1P2P3P4	10 bits
0x09	Subsampling	2	Single	0x33	10 → 8	P4	8 bits
0x0A			Single	0x34	—	P4	10 bits
0x0B		4	Single	0x33	10 → 8	P4	8 bits
0x0C			Single	0x34	—	P4	10 bits
0x0D		32	Single	0x33	10 → 8	P4	8 bits
0x0E			Single	0x34	—	P4	10 bits

10.1.2 Rolling shutter configuration

In this configuration, the two rows of the ADCs simultaneously feed two different paths of the video pipe. The ADCs do this respectively by the long path (longest integration time) and the short path (shortest integration time). The two paths are then merged into a single high dynamic range stream.

Figure 16. **Image processing for rolling shutter**

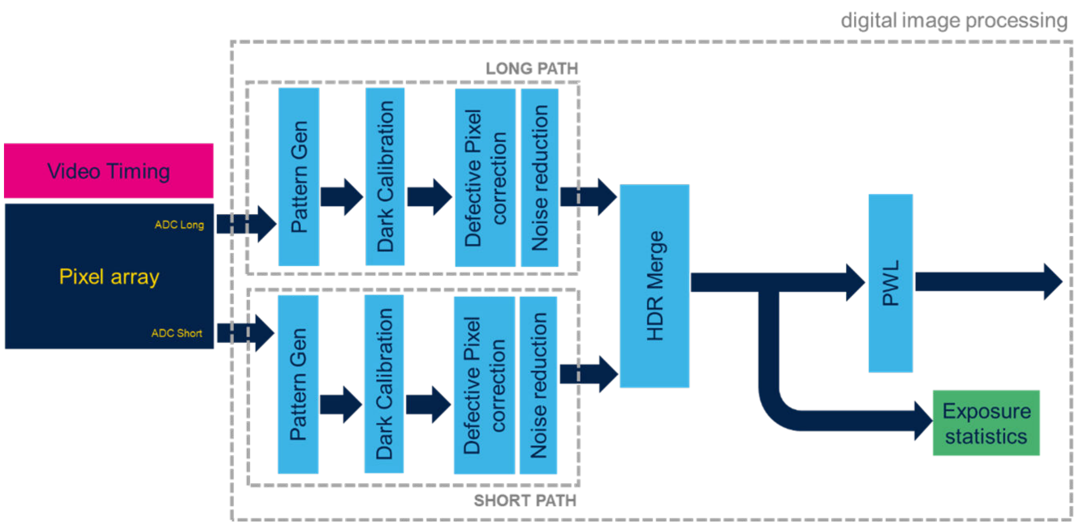


Table 37. **Rolling shutter configuration table**

Sensor configuration	Video timing configuration	Image processing configuration		Image output	
	VT mode	Data type format	PWL	Pixel format	Pixel dynamic
0x1A	SDR	0x2A	12 → 8	P1P2P3P4	8 bits
0x1B		0x2B	12 → 10	P1P2P3P4	10 bits
0x1C		0x2C	—	P1P2P3P4	12 bits
0x20	HDR	0x2B	18 → 10	P1P2P3P4	10 bits
0x21		0x2C	18 → 12	P1P2P3P4	12 bits

10.2 Orientation

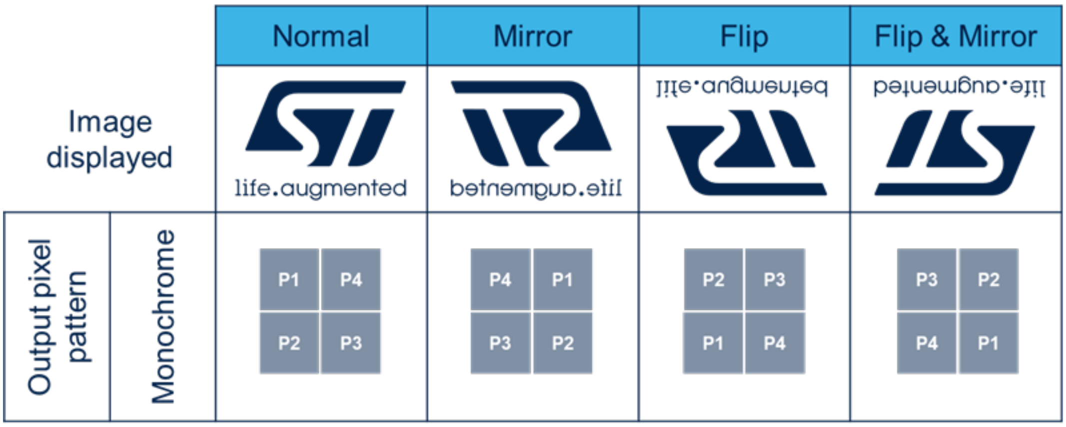
The device orientation is set with the ORIENTATION register from the STREAM_STATICS group.

Table 38. ORIENTATION [7:0]

7	6	5	4	3	2	1	0
Reserved						Flip	Mirror

The figure below presents the results of the different orientation options.

Figure 17. Image orientation



10.3 Region of interest

Four regions of interest are available: ROI_A, ROI_B, ROI_C, and ROI_D.

Table 39. ROI_SELECTION [7:0]

7	6	5	4	3	2	1	0
Reserved				ROI_D	ROI_C	ROI_B	ROI_A

Only one ROI can be selected per context. Each ROI has a set of four registers to configure it. These registers belong to the STREAM_STATICS group. The very first pixel has the coordinate (0, 0).

Table 40. List of states for ROI_SELECTION

VALUE	ROI_x
0	Disable
1	Enable

The ROI parameters must respect the following rules:

- The maximum width is 2560 and the maximum height is 1984.
- Offset and image size **in both directions must be multiples of 4.**
- WIDTH_OFFSET + WIDTH is ≤ 2560.
- HEIGHT_OFFSET + HEIGHT is ≤ 1984.

Figure 18. ROI definition

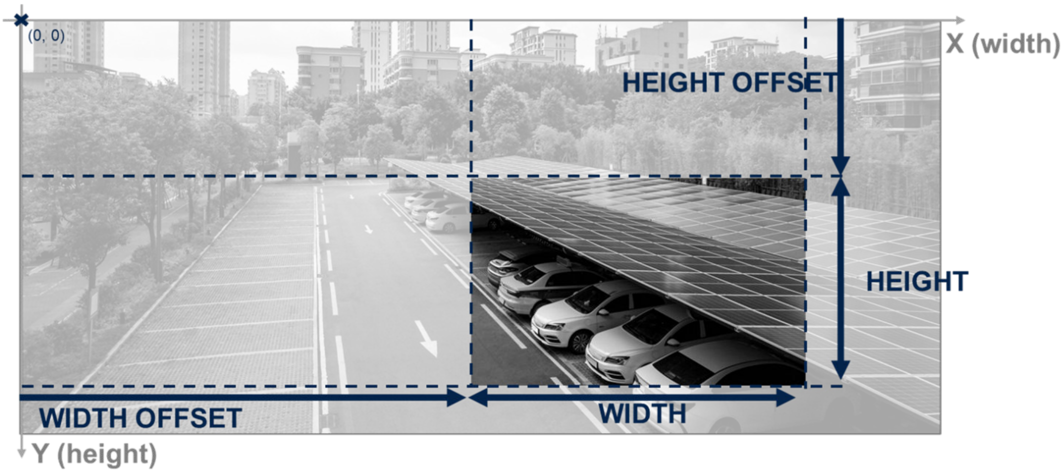


Table 41. ROI_x_WIDTH_OFFSET [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE															

Table 42. ROI_x_HEIGHT_OFFSET [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE															

Table 43. ROI_x_WIDTH [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE															

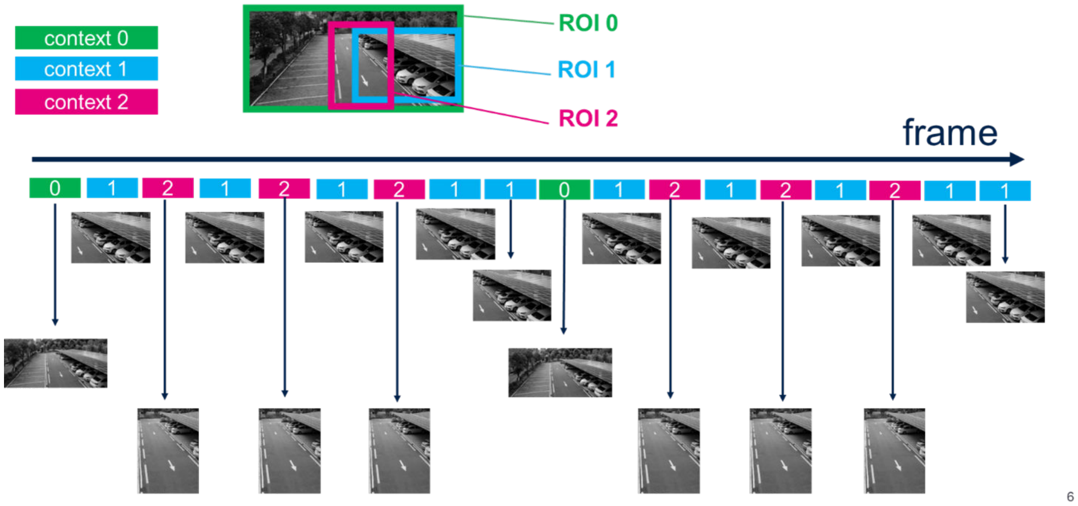
Table 44. ROI_x_HEIGHT [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE															

Example

Below is the ROI configuration for a multicontext use case.

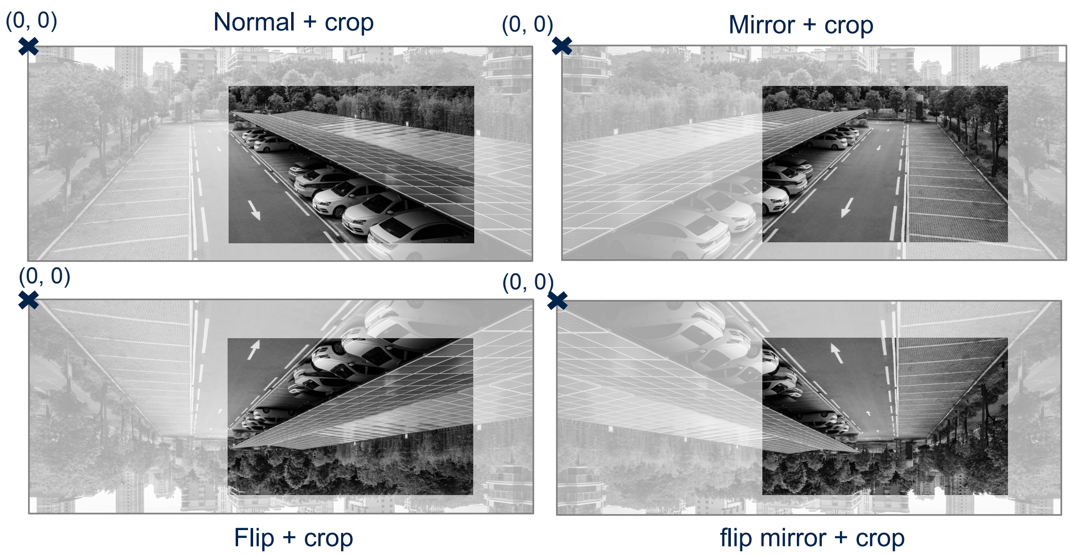
Figure 19. ROI and multicontexts



Orientation and ROI

When both orientation and ROI features are simultaneously enabled, the sensor performs first the orientation, then the crop. In the example below, the ROI configuration is the same while the orientation is switched.

Figure 20. ROI and orientation



10.4 Pattern generator

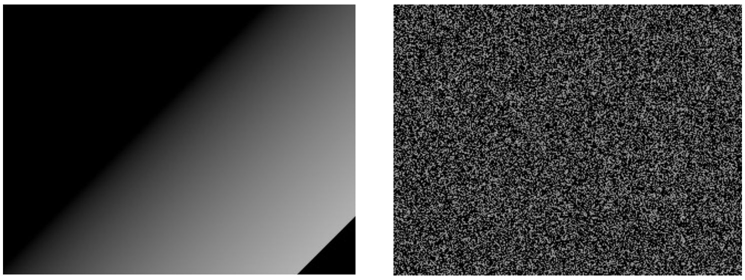
The devices can generate patterns for debugging purposes. The active pixel values are overridden when the register PATGEN_CTRL is other than 0.

Pattern generator control belongs to the STREAM_STATICS group. This feature cannot be enabled or disabled while streaming is ongoing.

The pattern generator allows the generation of digital patterns in the output frame. Available patterns are:

- Diagonal gray scale
- Pseudo-random data

Figure 21. Available patterns



Diagonal grayscale Pseudo-random data

Table 45. PATGEN_CTRL [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE								Reserved						ENABLE	

Table 46. PATGEN_CTRL ENABLE

ENABLE	Name	Description
0	PATGEN_BYPASS	Pattern generator is disabled
1	PATGEN_NORMAL	Pattern generator is enabled

Table 47. PATGEN_CTRL TYPE

TYPE	Name	Description
0x22	DGREY	Diagonal gray scale pattern selection
0x28	PN28	Pseudo random PN28 pattern selection

To generate an accurate pattern:

- DARKCAL_ENABLE must be set to disable (otherwise most of the image is black)
- DUSTER_CTRL must be set to disable to ensure the expected pixel to pixel stepping (pixel defect correction and noise reduction must be off)

If DARKCAL_ENABLE is kept to enable, the programmed pedestal value is added to the image; otherwise the field has no effect.

Refer to Section 10.5.1: Dark calibration bypass to control DARKCAL_ENABLE.

The pattern generator is RAW12 based. A rollover occurs when the counter goes over 4095.

The visible image is surrounded by a ring of two pixels that are part of the PATGEN count.

When PATGEN is configured as DGREY type, the first pixel has the value:

- 4 in rolling shutter RAW12 format
- 1 in global shutter RAW10 format (a decimation by 4 is applied)

If the configuration of the sensor enables a PWL compression, then the PATGEN output image is compressed. Refer to Table 36. Global shutter configuration table and Table 37. Rolling shutter configuration table.

10.5 Dark noise calibration

Because dark noise is dependent on the analog gain, and temperature, it is mandatory to track the dark noise level on a frame basis. The devices embed an automatic dark noise calibration mechanism. The calibration of the dark noise level is based on eight dark lines read ahead of the image. Dark pixels are processed independently according to four channels. Each channel is related to a position of the pixels in the matrix. The four channels are identified as P1, P2, P3, and P4.

Input dynamic range

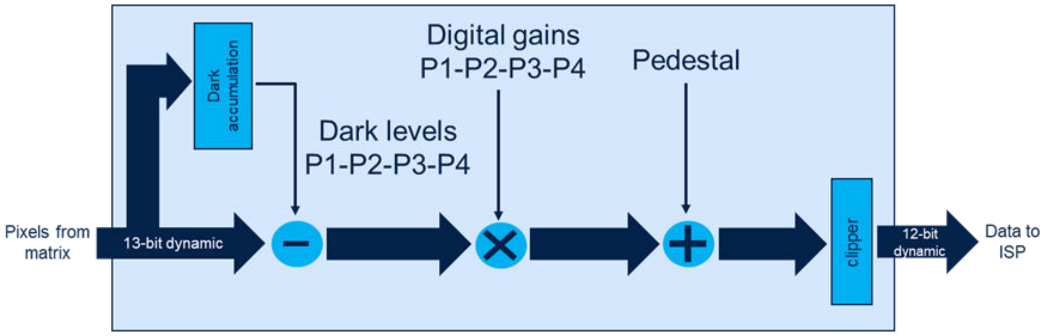
The dynamic range of input data is always 13 bits:

- For the rolling shutter, this is directly the ADC output.
- For the global shutter, the ADC output is shifted by 2 bits and aligned on the MSB (from 11 bits to 13 bits).

Output dynamic range

- The data are clipped at the output of the block to match exactly a 12-bit dynamic range (global shutter data remain shifted by 2 bits).

Figure 22. **Dark noise calibration**



10.5.1 Dark calibration bypass

The dark calibration mechanism can be bypassed using the register DARKCAL_ENABLE. This also bypasses the pedestal and digital gain control.

Dark calibration shall be bypassed when running pattern generator mode.

Table 48. **DARKCAL_ENABLE [7:0]**

7	6	5	4	3	2	1	0
DARKCAL_ENABLE							

Table 49. **DARKCAL_ENABLE value**

DARK_CAL_ENABLE	Dark calibration state	Description
0	DISABLE	DARKCAL is disabled
1	ENABLE	DARKCAL is enabled

10.5.2 Control

Each context has a register to control the dark noise calibration. This register belongs to the STREAM_CTXx_STATIC group. The control is applied simultaneously on both long and short paths.

Table 50. **DARKCAL_CTRL [7:0]**

7	6	5	4	3	2	1	0
Reserved						VALUE	

Table 51. **DARKCAL_CTRL descriptions**

Value	Name	Description
0	INDEPENDENT_CHANNELS	Data of each channel (P1/P2/P3/P4) are accumulated on a specific accumulator. The dark noise calibration is performed independently for each channel. This is the default mode for GS and RS.
1	GLOBAL_AVG	Data of each channel (P1/P2/P3/P4) are accumulated on a specific accumulator, but the accumulators are eventually merged. The dark noise calibration is performed with a global value on all channels. This mode is useful in subsampling or very low light (high noise).
2	BYPASS_DARK_AVG	In this mode, the accumulator values are not removed from the pixel data. This mode is for debug or evaluation.

10.5.3 Image pedestal

Each context has a register to control the pedestal. This register belongs to the STREAM_CTXx_STATIC group. The value is applied simultaneously on both long and short paths. The value is applied after the digital gains (see Section 11.4: Digital gains) to preserve the darkest part of the image.

Table 52. **DARKCAL_PEDESTAL [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				VALUE											

The DARKCAL_PEDESTAL default value is 0x0020 (32), but it is not applied for HDR rolling shutter configuration.

The DARKCAL_PEDESTAL value is directly applied on frames for SDR rolling shutter configurations. They must not exceed 16 for RAW8, and 64 for RAW10, otherwise the PWL compression reduces the expected pedestal.

The DARKCAL_PEDESTAL value is divided by four before being applied on frames for global shutter configurations. It must not exceed 64 for RAW8, otherwise the PWL compression reduces the expected pedestal.

10.5.4 Dark lines

A specific processing is applied on the pixels of the dark lines to prevent the computation of the dark noise accumulator to be corrupted by abnormal or leaky pixels. The block HDEFCOR is filtering out outliers. This block is enabled by default.

Table 53. **HDEFCOR_ENABLE [7:0]**

7	6	5	4	3	2	1	0
HDEFCOR_ENABLE							

Table 54. **HDEFCOR_ENABLE value**

HDEFCOR_ENABLE	Dark pixel correction state	Description
0	DISABLE	HDEFCOR is disabled
1	ENABLE	HDEFCOR is enabled

10.6 Defect correction and noise reduction

10.6.1 Pixel defect correction

The VB5943 and VD5943 sensors embed an algorithm that automatically corrects defective pixels. Singlet or couplet defective pixels are dynamically corrected on-the-fly by an exposure, analog gain and temperature auto-adaptive algorithm. The pixel matrix of the sensors are exempt of triplets. Triplets are considered as a texture in the image and are not corrected.

The figures below are examples of pixels detected as defective (white in illustration) which are corrected by the algorithm.

Figure 23. **Singlet and doublet correction**

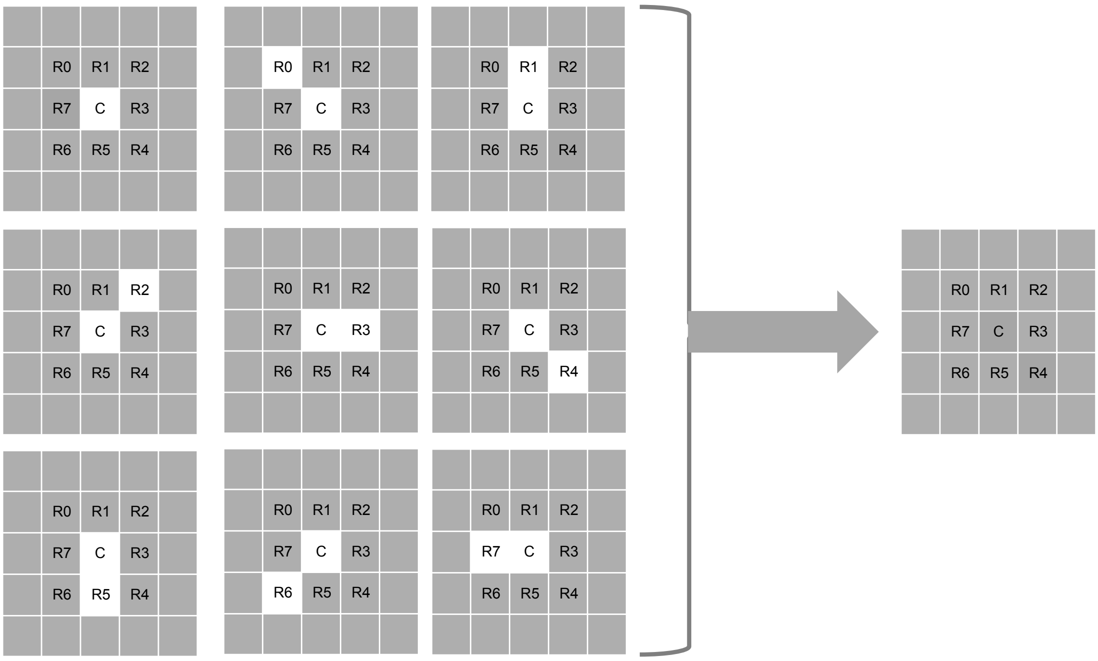
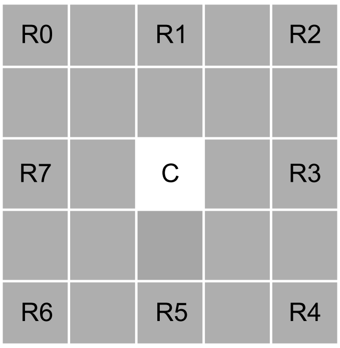


Figure 24. **Global shutter split exposure ring**



For global shutter single exposure or rolling shutter configurations, the correction algorithm uses a 3x3 kernel of pixels that surrounds the pixel to be corrected. P1/P2/P3/P4 pixels are considered as belonging to a common color plane.

For global shutter split exposure configuration, the correction algorithm uses a 5x5 kernel of pixels that surrounds the pixel to be corrected. P1 /P2/P3/P4 pixels are considered as belonging to different color planes.

Refer to Table 36. Global shutter configuration table and Table 37. Rolling shutter configuration table for existing configurations.

Each context has a register to configure defective pixel correction. This register, DUSTER_CTRL, belongs to the STREAM_CTXx_STATIC group.

Table 55. DUSTER_CTRL [7:0]

7	6	5	4	3	2	1	0
Reserved				NOISE_CORRECTION	Reserved	DYN_ENABLE	

Table 56. DYN_ENABLE values

DYN_ENABLE	Dynamic correction state	Description
0	DISABLE	Pixel correction is disabled
1	SINGLET and COUPLET correction	SINGLETs and COUPLETs are corrected
2	COUPLET correction only	Only COUPLETs are corrected

Each context has a register to configure the strength of the pixel defect correction. This register, DUSTER_DEF_COR_RATIO, belongs to the STREAM_CTXx_DYNAMIC group. A default strength is predefined with a corresponding ratio of 100%. This ratio can be adjusted in the range 50% (upper pixel defect correction strength) up to 200% (lower pixel defect correction strength).

Table 57. DUSTER_DEF_COR_RATIO [7:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DUSTER_DEF_COR_RATIO							

10.6.2 Pixel noise reduction

Pixel noise reduction is based on normal distribution law with adaptive filter depending on exposure and analog gain. The goal of the Gaussian filter is to reduce the noise within the image through weighted coefficients applied at each pixel within a kernel of 5x5 pixels.

Each context has a register to enable or disable the pixel noise reduction. This register, DUSTER_CTRL, belongs to the STREAM_CTXx_STATIC group.

Table 58. NOISE_CORRECTION

NOISE_CORRECTION	Noise reduction state	Description
0	DISABLE	Noise reduction is disabled
1	ENABLE	Noise reduction is enabled

Then DUSTER_NOISE_RED_RATIO allows to set the wanted global strength of the noise reduction. The final result of the noise reduction is a mix of the original pixel value and the corrected pixel.

Each context has its own register DUSTER_NOISE_RED_RATIO belonging to the STREAM_CTXx_DYNAMIC group.

A default strength is predefined with a corresponding ratio of 100%. This ratio can be adjusted in the range 50% (lower noise reduction) up to 200% (upper noise reduction).

Table 59. **DUSTER_NOISE_RED_RATIO [7:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DUSTER_NOISE_RED_RATIO							

The noise-corrected output pixel is a mix between the input pixel and the Gaussian filtered pixel.

The register DUSTER_NOISE_REDUC_BLEND is defining the proportion of the blending.

Each context has its own register DUSTER_NOISE_REDUC_BLEND belonging to the STREAM_CTXx_STATIC group.

The blending can be adjusted in the range 16 (output pixel value is fully coming from Gaussian noise reduction algorithm) down to 0 (output pixel value is equal to input pixel value).

The default value is 16.

Table 60. **DUSTER_NOISE_REDUC_BLEND [3:0]**

7	6	5	4	3	2	1	0
Reserved				DUSTER_NOISE_REDUC_BLEND			

10.7 HDR operation

An HDR picture is obtained by merging two captures of the same scene using two different exposure periods. A long exposure period capture provides additional details of low-light zones while a short exposure period capture provides additional details of high-light zones.

The long exposure is made first, followed a few lines later by the short exposure producing a quasi-simultaneous image capture. The SHORT exposure cannot exceed 396 lines.

The maximum ratio of the exposure duration is defined by the architecture of the digital pipe of the sensor. The pixel conversion is 12 bits and the ISP pipe is 18 bits. The maximum exposure ratio is then 2^(18-12) that is, 64.

A systematic PWL compression is applied to the HDR image in order to fit the 18-bit image dynamic inside a RAW10 or RAW12 output format.

10.7.1 HDR merge

The host must provide:

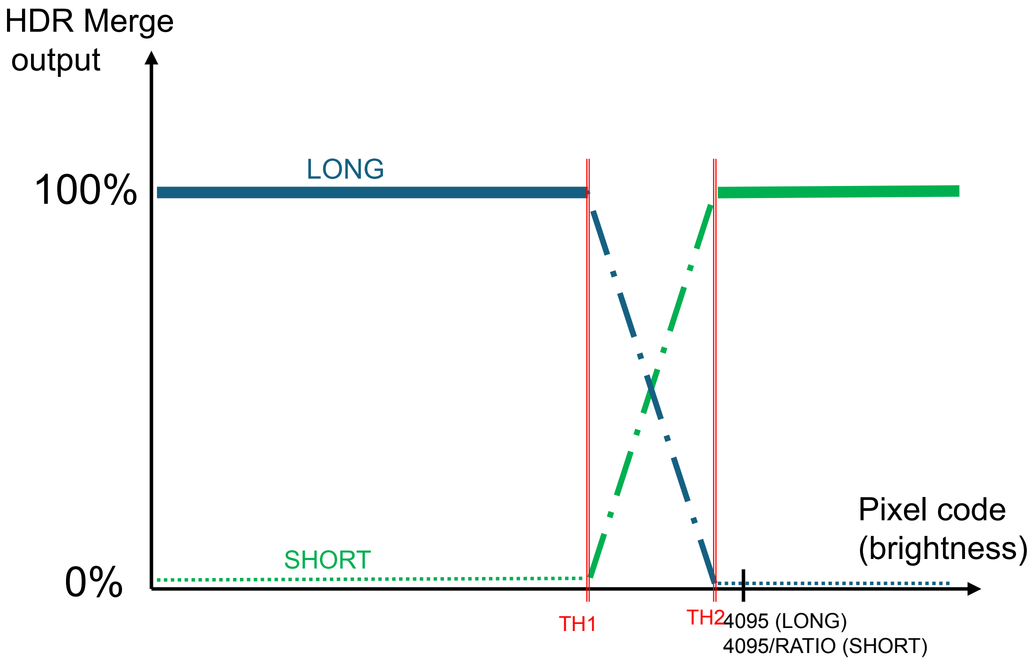
- LONG exposure time in register INTEGRATION_TIME_PRIMARY[15:0]
- SHORT exposure time in register INTEGRATION_TIME_SHORT[15:0]

Each context has its own register INTEGRATION_TIME_PRIMARY and INTEGRATION_TIME_SHORT belonging to the STREAM_CTXx_DYNAMIC group.

Based on the ratio of long to short integration time, the embedded running firmware is computing the required HDR gains to build the reconstructed HDR image.

Because pixels from LONG exposure capture have a higher SNR than SHORT exposure ones, the HDR merger outputs LONG exposure information as long as it is not saturated. When pixels from LONG exposure reach a level close to the maximum code (4095), the SHORT exposure information is progressively injected into the HDR merger until pixels from LONG exposure are near to saturate. At that point only pixels SHORT exposure information is kept. This HDR merge transition occurs in the upper part of the LONG exposure dynamic.

Figure 25. HDR merge



10.7.2 Ghost removal

The HDR image is obtained by combining two consecutive captures delayed by the short exposure time. Nevertheless, this small amount of time can be enough to produce a ghost effect during HDR construction when motion is present during the capture.

In the absence of ghost in the image, long and short exposure pixel information multiplied by their respective HDR gains should be similar. If the difference is greater than a given threshold, a ghost artifact is detected and the ghost algorithm outputs short pixel information which contains less blur than the long one.

When a ghost is detected, the ghost removal action has priority on the HDR merge.

Table 61. HDR_GHOST_FLICKER_CTRL [7:0]

7	6	5	4	3	2	1	0
Reserved						VALUE	

Table 62. List of states for HDR_GHOST_FLICKER_CTRL

Value	Name	Description
0	DISABLE	Normal HDR merging
1	GHOST_REMOVAL	Ghost removal enabled

10.8 PWL

The role of the PWL module is to limit pixel dynamic to decrease data bandwidth before CSI-2 transmission.

The VB5943 and VD5943 sensors have five PWL curves which are selected depending on the general configuration (refer to Table 63. Default PWL curves). The curves are made of 32 segments and have been computed to optimize image quality. After companding and transmission, image data are supposed to be expanded by the receiver to retrieve the linearity of the dynamic.

Table 63. Default PWL curves

Shutter mode	Pixel dynamic	PWL
GS	SDR	10 bits to 8 bits
RS	SDR	12 bits to 8 bits
		12 bits to 10 bits
	HDR	18 bits to 10 bits
		18 bits to 12 bits

Figure 26. PWL curves

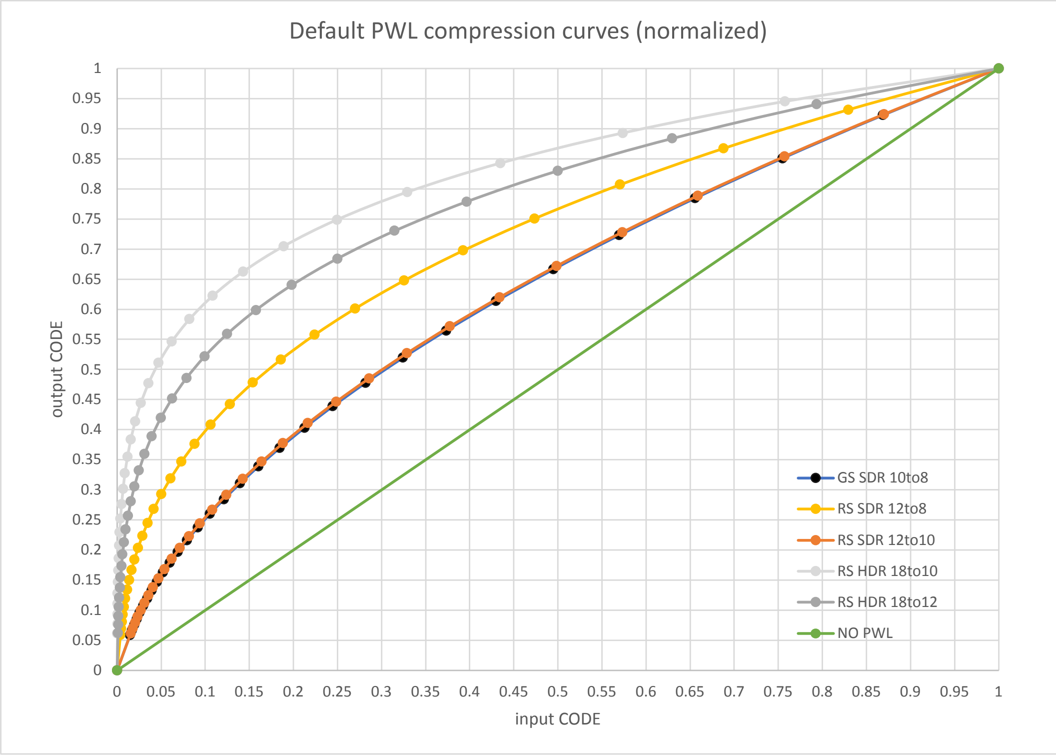


Table 64. Default PWL knee coordinates

	SDR						HDR			
	RAW10->8		RAW12->10		RAW12->8		RAW18->12		RAW18->10	
Knee	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
0	0	0	0	0	0	0	0	0	0	0
1	15	15	63	63	15	15	255	255	63	63
2	17	17	72	71	18	17	321	310	83	78
3	19	19	83	81	21	20	405	368	110	95
4	22	21	95	91	26	23	510	430	145	113
5	26	24	109	102	31	27	643	494	191	131
6	30	27	126	114	38	30	810	562	253	150
7	34	30	145	127	46	34	1021	634	334	169
8	40	33	166	141	55	38	1286	709	440	190
9	46	37	191	156	66	42	1621	788	581	212

	SDR						HDR			
	RAW10->8		RAW12->10		RAW12->8		RAW18->12		RAW18->10	
Knee	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output
10	53	41	220	172	80	47	2042	872	768	234
11	61	45	253	189	97	52	2574	959	1013	258
12	70	50	291	208	117	57	3243	1052	1338	282
13	81	55	334	228	141	62	4086	1149	1766	308
14	93	60	384	249	170	68	5149	1251	2332	335
15	107	66	442	273	205	74	6488	1359	3078	363
16	123	72	507	298	247	81	8176	1472	4064	392
17	142	79	583	325	298	88	10302	1592	5365	423
18	164	86	671	355	360	96	12982	1717	7083	455
19	189	94	771	386	434	104	16358	1849	9351	488
20	217	103	886	420	523	113	20613	1989	12345	523
21	250	112	1018	457	631	122	25974	2135	16298	559
22	288	122	1170	496	761	132	32730	2289	21517	597
23	332	132	1345	539	917	142	41242	2451	28407	637
24	382	144	1546	585	1106	153	51968	2622	37502	678
25	440	157	1777	634	1333	165	65485	2801	49511	721
26	506	170	2042	688	1608	178	82516	2991	65364	766
27	583	185	2347	745	1938	192	103978	3190	86294	813
28	671	200	2698	807	2337	206	131021	3399	113925	863
29	772	217	3100	874	2818	222	165097	3619	150404	914
30	889	236	3563	946	3397	238	208037	3851	198564	968
31	1023	255	4095	1023	4095	255	262143	4095	262143	1023

10.8.1 PWL selection

In addition to default curves, it is possible to program a custom curve in rolling shutter mode. Although this new curve is common to all contexts, the selection (default/custom) is achieved by using a context according to the PWL_SEL register (STREAM_CTXx_STATIC group).

Table 65. PWL_SEL [7:0]

7	6	5	4	3	2	1	0
Reserved							VALUE

Table 66. List of states for PWL_SEL

Value	Name	Description
0	DEFAULT_LUT	The default curve is used
1	CUSTOM_LUT	The custom curve is used

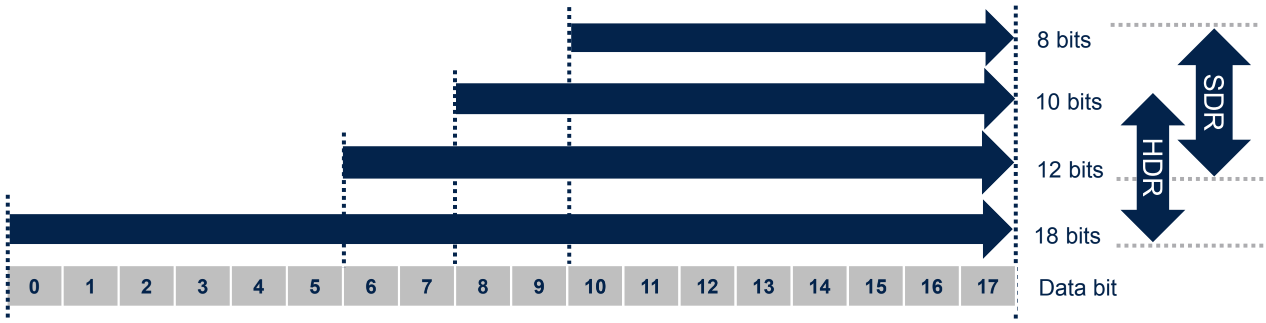
10.8.2 Programming the custom curve

Custom PWL only applies for rolling shutter configuration.

When using a custom PWL and rolling shutter SDR configuration, the first transmitted frame is discarded.

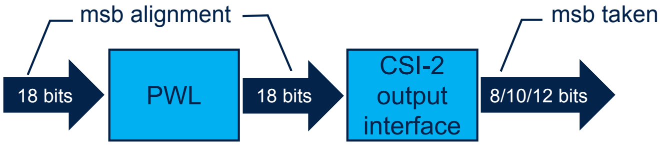
Both the input and output of the PWL module have a unique range of 18 bits. Data is aligned to the MSB to cope with the different pixel dynamics (see figure below).

Figure 27. Alignment to the MSB



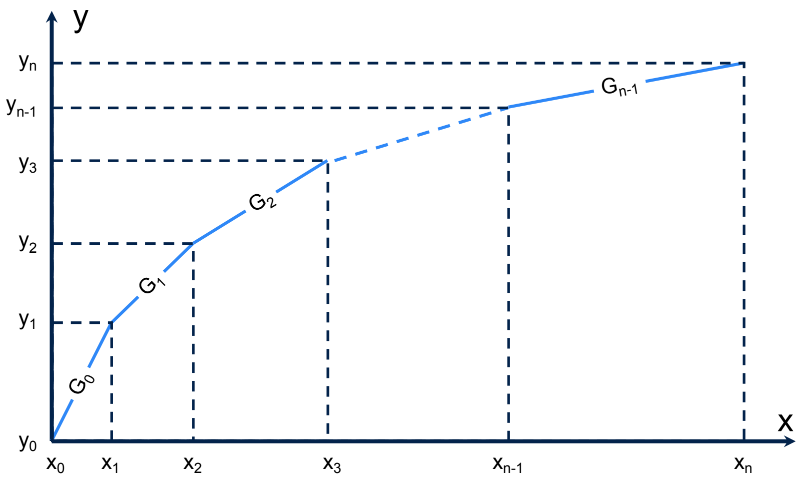
The CSI-2 output interface selects the relevant bits automatically according to the sensor's configuration (see figure below).

Figure 28. Bit selections of the PWL



The PWL curve is characterized by the start point coordinates (X_n, Y_n) for each segment, the segment slope gradient (G_n) and the exponent bias. Note that coordinates (X_n, Y_n) are called "knee points". The 32 points describing the PWL are programmable.

Figure 29. PWL compression example



The PWL formula is:

$$Y = Y_n + (X - X_n) \times G_n$$

For X in [X_n, X_{n+1}].

Where:

- X is the pixel input value.
- Y is the pixel output value.
- X_n is the start point abscissa of the selected segment.
- Y_n is the start point ordinate of the selected segment.
- G_n is the gradient (slope) of the selected segment.

G_n is represented in the normalized mantissa/exponent format by the equation:

$$G_n = \{leading\ bit, mantissa\} \times 2^{global\ exponent}$$

As the mantissa is an 8-bit field, the PWL gradient can be expressed as follows in fixed point:

$$G_n = \{256 + mantissa\} \times 2^{\text{exponent} - bias}$$

The leading bit (256) is not stored in the memory. Its default value is 256 except when both the exponent and mantissa values are zero. This is a convention of the gradient zero coding.

The global exponent is represented with a bias value subtracted from the PWL_GRADIENT exponent.

The PWL_GRADIENT exponent is a 4-bit field. The bias is a 5-bit field parameter that is constant for the whole shape written in the field PWL_EXPO_BIAS.

There are four groups of registers to program the PWL in the STREAM_STATICS group. They are:

- 32 PWL_ABSCISSA_n 32-bit registers
- 32 PWL_ORDINATE_n 32-bit registers
- 32 PWL_GRADIENT_n 32-bit registers
- 1 PWL_EXPO_BIAS register

As the ISP pipe is 18 bits wide, X_n (the input path of the PWL stage) and Y_n (the output path of the PWL stage) can have values from 0 to 262143.

The X_n coordinates are stored in the lower 18 bits of the PWL_ABSCISSA_n field.

The Y_n coordinates are stored in the lower 18 bits of the PWL_ORDINATE_n field.

The G_n slope is stored in the lower 12 bits of the PWL_GRADIENT_n field, which is composed of the exponent and mantissa.

Table 67. PWL_ABSCISSA_n [31:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														VALUE																	

Table 68. PWL_ORDINATE_n [31:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														VALUE																	

Table 69. PWL_GRADIENT_n [31:0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				Exponent				Mantissa							

Table 70. PWL_EXPO_BIAS [7:0]

7	6	5	4	3	2	1	0
Reserved			VALUE				

Details regarding the calculations

- The first coordinates must be (0, 0).
- Output values are rounded to the nearest integer.
- Output values are clipped to the maximum value of the curve.

10.8.3 PWL status

Information regarding the PWL module is available in the STATUS register group.

Table 71. PWL [7:0]

7	6	5	4	3	2	1	0
Reserved						LUT_SEL	ENABLE

Table 72. LUT_SEL

LUT_SEL	Name	Description
0	DEFAULT_LUT	The default curve is used
1	CUSTOM_LUT	The custom curve is used

Table 73. ENABLE

ENABLE	Name	Description
0	DISABLE	Data are not companded. Nominal pixel dynamic
1	ENABLE	Data are companded

10.9 Exposure statistics

Table 74. EXPO_STAT_SELECTION [7:0]

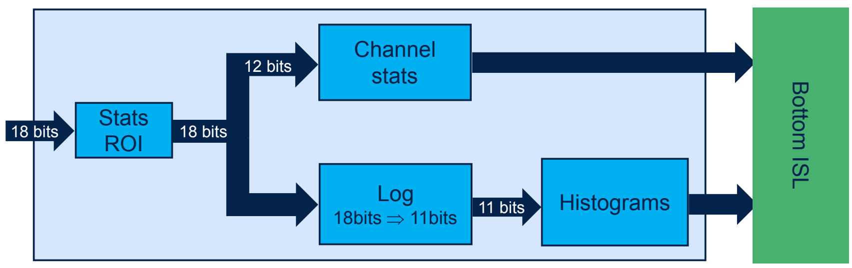
7	6	5	4	3	2	1	0
Reserved							0

Exposure statistics data are available to the host to analyze the content of the image. Two modules are available for collecting exposure statistics.

- Channel statistics provide simple measures over 16 zones of the image
- Histograms provide complete histogram of the image

The input data has a unique dynamic range of 18 bits, assuming 10-bit/12-bit video stream is shifted to the MSB.

Figure 30. Overview of the exposure statistics



Exposure statistics are transmitted embedded in the video stream to the host. These data are sent as meta data at the end of the frame (bottom status lines). Refer to Section 12.2.5: Bottom status lines for a complete description of the status lines.

The EXPO_STAT_CTRL register controls the content of the bottom status lines. It is available in the STREAM_STATICS group.

Table 75. **EXPO_STAT_CTRL [7:0]**

7	6	5	4	3	2	1	0
Reserved						VALUE	

Table 76. **List of states for EXPO_STAT_CTRL**

VALUE	Name	Description
0	DISABLE	No statistics are transmitted
1	CHANNEL_STATS_ONLY	Only channel statistics are transmitted
2	CHANNEL_STATS_2_HISTOGRAMS	Channel statistics and two histograms are transmitted
3	CHANNEL_STATS_3_HISTOGRAMS	Channel statistics and three histograms are transmitted

10.9.1 ROI of exposure statistics (STAT ROI)

Exposure statistics are collected in an ROI controlled by a set of registers in each context (STREAM_CTXx_STATIC group).

The ROI of the exposure statistics must be wholly included within the image. Assuming that WIDTH and HEIGTH are the image resolutions defined in Section 10.3: Region of interest, the parameters of the STAT ROI must respect the rules below:

- Offset in both directions must be a multiple of 4.
- STAT ROI size in both directions must be a multiple of 16.
- STAT_ROI_WIDTH_OFFSET + STAT_ROI_WIDTH ≤ WIDTH
- STAT_ROI_HEIGTH_OFFSET + STAT_ROI_HEIGHT ≤ HEIGTH

Figure 31. **ROI of the exposure statistics**

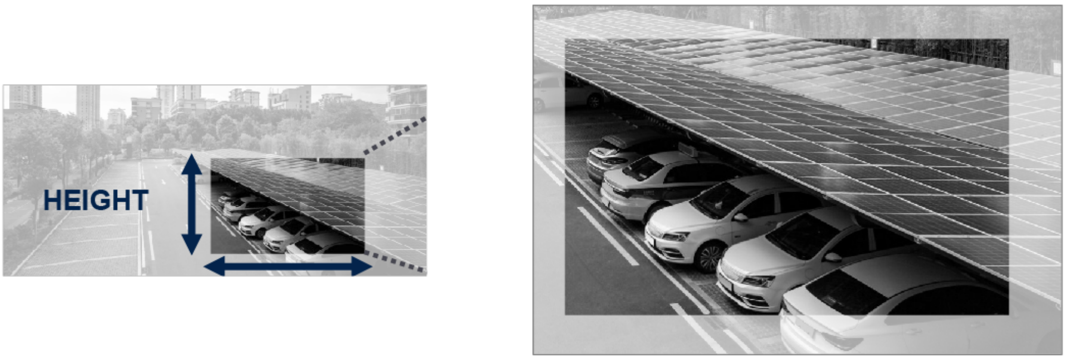


Table 77. **STAT_ROI_WIDTH_OFFSET [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value															

Table 78. **STAT_ROI_WIDTH [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value															

Table 79. **STAT_ROI_HEIGHT_OFFSET [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value															

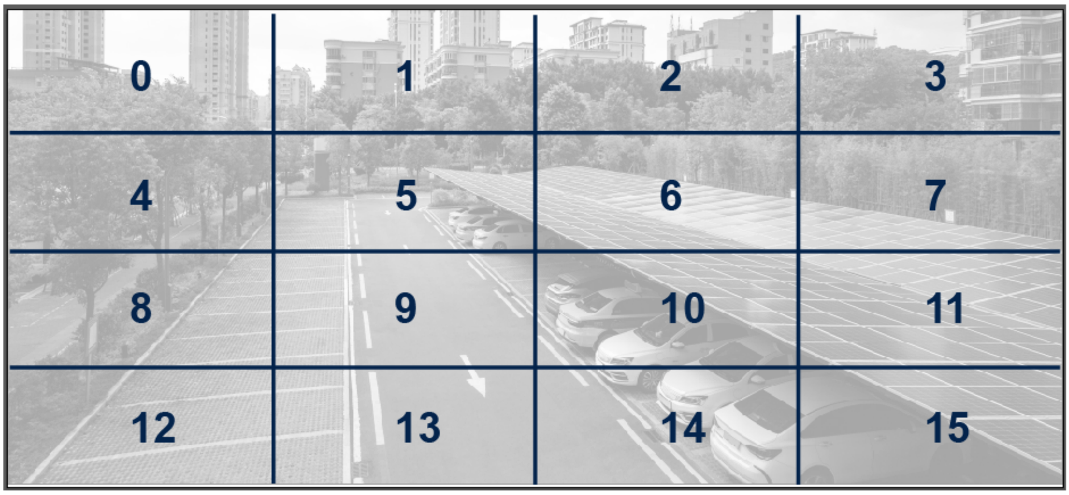
Table 80. STAT_ROI_HEIGHT [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value															

10.9.2 Channel statistics

This module provides statistics for 16 identical zones of the STAT ROI described in the figure below. Each zone has three independent accumulators to accumulate pixel values. The module allows specific information for two possible exposures in the matrix. These exposures are single or split exposure modes. The number of pixels collected for the statistics and all the accumulator final values is available on a frame basis in the bottom status lines (refer to Section 12.2.5: Bottom status lines). Only the 12-bit MSBs are considered for the accumulation.

Figure 32. Channel stats zones



The CHANNEL_STAT_CFG register controls the association of the three groups of accumulators with the pixel position. The accumulators are different depending on the output pattern. Pixels are accumulated in four different banks inside a 4x4 kernel. The banks are green, purple, yellow, and red (see figure below). Five configurations of the accumulator are available.

Figure 33. Pattern and channel for channel statistics module

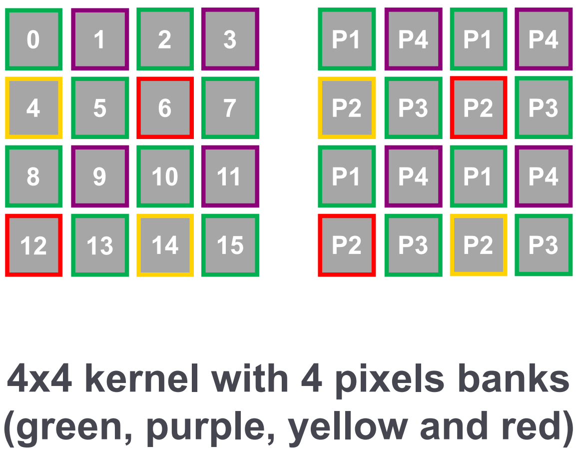


Table 81. CHANNEL_STAT_CFG [7:0]

7	6	5	4	3	2	1	0
CONFIG							

Statistics are computed in four different banks inside a 4x4 kernel. Three accumulators can be output depending on the CHANNEL_STAT_CFG value.

Table 82. List of calculated accumulators

CHANNEL_STAT_CFG	Accumulator 0	Accumulator 1	Accumulator 2
0	Green bank (P1)	Green bank (P3)	Purple bank (P4)
1	P1 position 0	P2 position 12	P4 position 11
2	P1 position 0	P2 position 14	P4 position 3
3	P2 position 12	P2 position 4	P3 position 7
4	P2 position 12	P2 position 4	P4 position 11

Output data are composed of:

- 48 accumulators: Three groups of 16 accumulators.
- Three pixel counters: One per group because the same number of pixels is accumulated in each zone.

The accumulators and pixel counters have the respective structure below.

Table 83. STAT_ACCx_ZONE_y [31:0]

Group x = [0 : 2] and zone y = [0 : 15]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUM																															

Note: The STAT_NBP provides the number of pixels used for performing the accumulation of one zone and one group of pixels. Refer to Table 82. List of calculated accumulators to identify the number of pixels from a macro block that contribute to the computation.

Table 84. STAT_NBPx [31:0]

Channel x = [0 : 2]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUM																															

In the bottom status lines (refer to Section 12.2.5: Bottom status lines), the channel statistics data are organized as shown below.

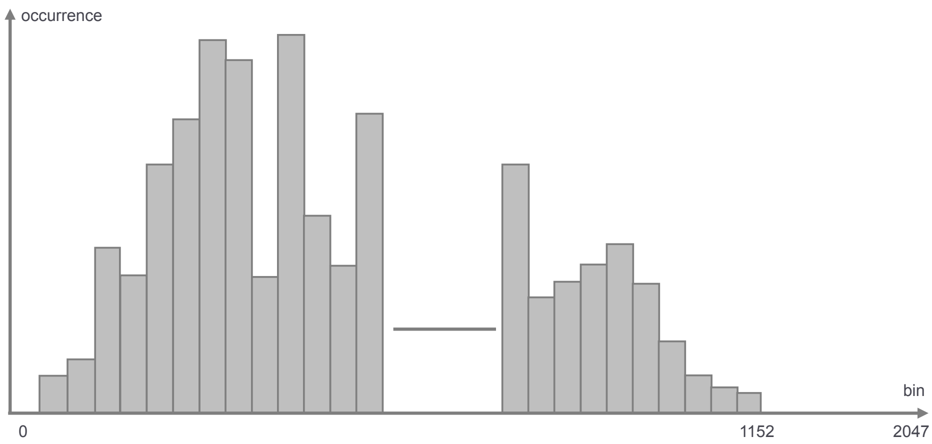
Table 85. Channel statistics data organization

Bytes	Data
[63:0]	STAT_ACC0_ZONE_0 to ZONE_15
[127:64]	STAT_ACC1_ZONE_0 to ZONE_15
[191:128]	STAT_ACC2_ZONE_0 to ZONE_15
[195:192]	STAT_NBP0
[199:196]	STAT_NBP1
[203:200]	STAT_NBP2

10.9.3 Histograms

The histogram module provides the distribution of pixel values for the four different pixel banks. Three sets of memories are available to store histograms of three different banks. Histograms are available on a frame basis in the bottom status lines (refer to Section 12.2.5: Bottom status lines).

Figure 34. Histogram overview



The original 18-bit data stream is processed by a logarithmic module to shift the dynamic from 18 to 11 bits. Each bin is stored with a 21-bit memory element. The histograms have 2048 bins (2^{11}) to support the 11-bit dynamic range. But due to the logarithmic module calculation, the full dynamic cannot be greater than 1152.

The HISTO_CFG register controls the association of the bank with the histogram memory.

Figure 35. Bank pattern for histogram module

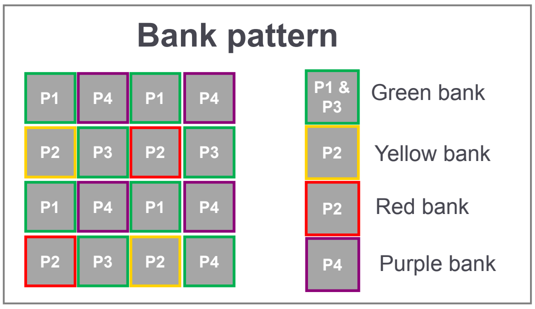


Table 86. HISTO_CFG [7:0]

7	6	5	4	3	2	1	0
CONFIG							

The EXPO_STAT_SELECTION must be set to 0. Histograms are computed as shown in the table below.

Table 87. List of histograms configurations

HISTO_CFG	HISTOGRAM 0	HISTOGRAM 1	HISTOGRAM 2
0	P1 + P3 pixels	P1 + P3 pixels	Purple bank
1	Red bank	P1 pixels	Yellow bank
2	Purple bank	P1 pixels	Red bank
3	Purple bank	P1 pixels	Yellow bank

All 2048 elements of the histogram have the following structure: **HISTOGRAM_x_BIN_y [31:0]** (channel x = [0:2] and element y = [0:2047]).

Table 88. **HISTOGRAM_x_BIN_y [31:0]**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OCCURENCE																				

Table 89. **Histogram output organization**

Histogram position	EXPO_STAT_CTRL			
	0	1	2	3
0-8191	X	X	HISTOGRAM_0	HISTOGRAM_0
8192-16383	X	X	HISTOGRAM_1	HISTOGRAM_1
16384-24575	X	X	X	HISTOGRAM_2

The organization of the histogram data in the bottom status line is described in section Section 12.2.5: Bottom status lines.

All 2048 elements of the histogram have the following structure: **HISTOGRAM_x_BIN_y [31:0]** (channel x = [0:2] and element y = [0:2047]).

Table 90. **HISTOGRAM_x_BIN_y [31:0]**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OCCURENCE																				

11 Exposure

The exposure is a combination of the following three parameters:

1. Integration time
2. Analog gain
3. Digital gain

A set of these exposure controls is available in each context (STREAM_CTXx_DYNAMIC groups). These values can be updated during the SW STANDBY or STREAMING states.

The synchronization of these parameters is described below. The sensor latches the parameters during the frame preceding the integration. According to the exact time of latch, the new parameters take effect with a latency of two or three frames.

Figure 36. Exposure parameters sampled before latch point

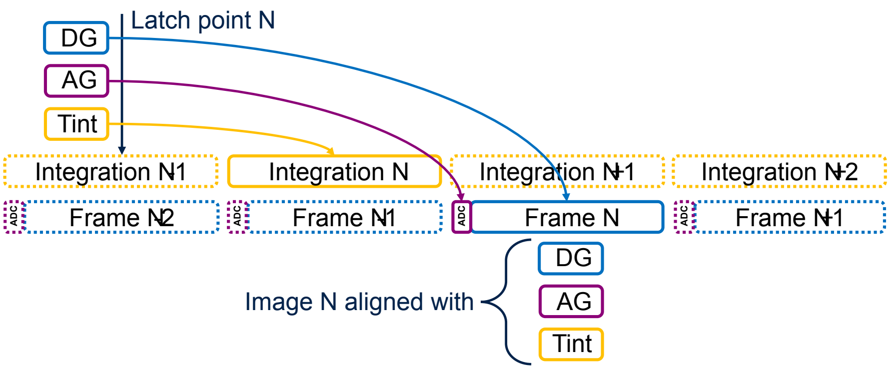
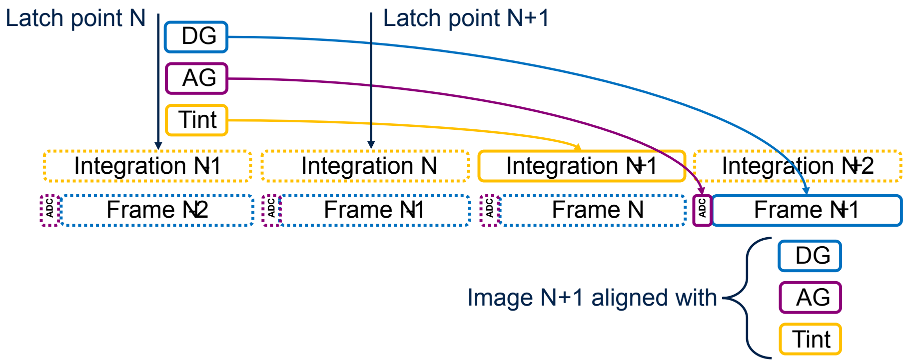


Figure 37. Exposure parameters sampled after latch point



Because the VB5943 and VD5943 sensor have several advanced features based on multiexposure, several registers are available to control the integration time.

All these registers are expressed in line time. If the line time is modified, the integration time is modified accordingly. The default line duration is 9 µs.

11.1 Integration time of the global shutter

The minimum integration time is scaled in number of lines, with a minimum of four lines.

This represents 36 µs with the minimum allowed line length.

Artifacts are present in the image if the integration time is set below four lines.

11.1.1 Single exposure

The INTEGRATION_TIME_PRIMARY register controls the integration time of all pixels. The GPIO STROBE signal applies to INTEGRATION_TIME_PRIMARY period.

11.1.2 Split exposure

The integration time of the P1P2P3 pixels is controlled by the INTEGRATION_TIME_PRIMARY register.

The integration time of the P4 pixels is controlled by the INTEGRATION_TIME_P4 register.

INTEGRATION_TIME_PRIMARY and INTEGRATION_TIME_P4 must be either equal to or have a difference greater than 10 lines.

A difference of 1 to 10 lines can introduce artifacts in the image.

The sensor manages the integration sequence to achieve INTEGRATION_TIME_P4 and INTEGRATION_TIME_PRIMARY ending at the same time.

The GPIO STROBE signal applies to the INTEGRATION_TIME_P4 period.

11.1.3 Subsampling

The INTEGRATION_TIME_PRIMARY register controls the integration time of all pixels. The GPIO STROBE signal applies to INTEGRATION_TIME_PRIMARY period.

11.1.4 Maximum exposure value

The maximum INTEGRATION_TIME_PRIMARY exposure value is limited to the programmed FRAME_LENGTH minus 28.

The maximum INTEGRATION_TIME_P4 exposure value is also limited to the programmed FRAME_LENGTH minus 28.

If the integration time is configured above this limit, the firmware clips the applied integration time and issues a warning.

Note: *In order to preserve the image quality at low frame rate (below 30 fps), the following additional I^2C instructions can be written in register addresses 0xa7e0 and 0xa7ec, to be applied after the START_STREAM command completes.*

```
Write8(0xa7e0, 0x0b);
Write8(0xa7ec, 0x34);
```

11.2 Integration time of the rolling shutter

In rolling shutter mode, the integration time starts 320 pixel clock cycles before the first integration line. So, the applied integration time is 320 pixel clock cycles longer than the programmed integration time in lines. This assuming a default pixel clock of 93.75 MHz. The corresponding integration time offset is 3.41 µs.

In rolling shutter mode, the minimum integration time is 1 line + 3.41 µs. There is no GPIO STROBE capability in rolling shutter mode.

11.2.1 Standard dynamic range

The integration time is controlled by the INTEGRATION_TIME_PRIMARY register.

11.2.2 High dynamic range

The INTEGRATION_TIME_PRIMARY register controls the long integration time.

The INTEGRATION_TIME_SHORT register controls the short integration time.

The short integration time is limited to 396 lines. If INTEGRATION_TIME_SHORT register is programmed above 396 lines, the sensor may stop transmitting images, without notification.

The short-to-long ratio should remain in the range [8–64].

11.2.3 Maximum exposure value

In SDR mode, the maximum INTEGRATION_TIME_PRIMARY exposure value is limited to the programmed FRAME_LENGTH minus 10. The INTEGRATION_TIME_SHORT must be set to 2.

In HDR mode, the maximum INTEGRATION_TIME_SHORT + INTEGRATION_TIME_PRIMARY exposure value is limited to the programmed FRAME_LENGTH minus 13.

In HDR mode the following equations must be respected:

$$FRAME\_LENGTH - IMAGE\_HEIGHT - INTEGRATION\_TIME\_SHORT > 62$$

$$FRAME\_LENGTH - INTEGRATION\_TIME\_SHORT - INTEGRATION\_TIME\_PRIMARY > 27$$

If the integration time is configured above this limit, the sensor may stop transmitting images, without notification.

11.3 Analog gain

The ANALOG_GAIN register applies the same analog gain on all pixels during the readout.

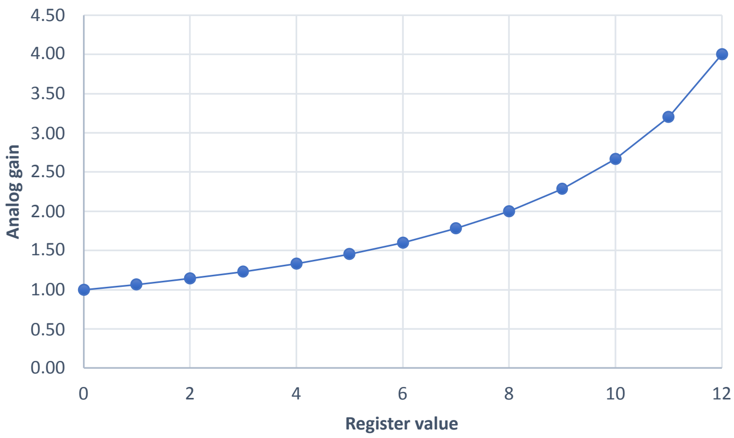
Table 91. **ANALOG_GAIN [7:0]**

7	6	5	4	3	2	1	0
Reserved				ANALOG_GAIN			

The gain is computed with the following formula: $Analog\ gain = \frac{16}{16 - ANALOG\_GAIN}$

The value of the register is limited to the range [0–12], which corresponds to an analog gain in the range [1.0–4.0].

Figure 38. **Control of the analog gain**



11.4 Digital gains

Digital gain is usually useful for controlling the global exposure. Four registers are available for controlling independently the digital gain of the four channels. They are applied in the dark calibration block at the input of the digital processing chain. Each register is coded as a fixed point of 5.8 (with a granularity of 1/256).

For example

- Register = 0x0100 = 256/256 = 1.0 (default value)
- Register = 0x0280 = 640/256 = 2.5

The register range is [0x0000, 0x1FFF] = [0.0, 31.99].

Table 92. **DIGITAL_GAIN_X [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			INTEGER					FRACTIONNAL							

The register DIGITAL_GAIN_P1 controls the digital gain of the P1 channel.

The register DIGITAL_GAIN_P2 controls the digital gain of the P2 channel.

The register DIGITAL_GAIN_P3 controls the digital gain of the P3 channel.

The register DIGITAL_GAIN_P4 controls the digital gain of the P4 channel.

The same digital gain value must be applied to all pixels (P1, P2, P3, and P4) when the sensor is running in rolling shutter, or in single exposure in global shutter.

A specific digital gain can be applied to the P4 pixel when the sensor is running in global shutter split exposure or in subsampling.

11.5 Group parameter hold

To synchronize the exposure settings (integration times and analog and digital gains), it is possible to use the group parameter hold (GPH) register. When this register is enabled, the UI exposure settings are not considered by the sensor. The previous settings remain active and control the exposure. When the GPH register is released, the sensor uses the UI exposure settings. In other words, the sensor automatically samples the value at the next latch point. This feature avoids the need for precise synchronization between the host and the sensor.

The GPH register is available for each context (STREAM_CTXx_DYNAMIC group).

Figure 39. **GPH with four contexts**

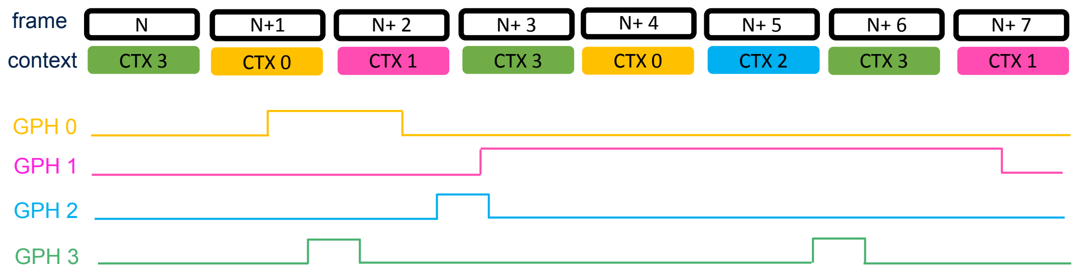


Table 93. **GPH [7:0]**

7	6	5	4	3	2	1	0
Reserved							HOLD

Table 94. **List of states for GPH**

Hold	Name	Description
0	DISABLE	Dynamic parameters are sampled by the sensor
1	ENABLE	Dynamic parameters are not sampled by the sensor

11.6 Exposure status

The three integration times, the common analog gain, and the four digital gains are updated on a frame basis in the STATUS register group.

12 Output interface

The output interface includes a quad-lane MIPI D-PHY interface that supports data rates of up to 6 Gbps (1.5 Gbps per lane). The minimum data rate is 600 Mbps per lane. But using the configuration rolling shutter and HDR, the minimum data rate is limited to eight times the pixel clock (that is, 750 MHz when the pixel clock is running to its default 93.75 MHz value).

There is a possibility to transfer data through the CSI-2 interface:

- Status lines
- SMIA RAW12, SMIA RAW10 or SMIA RAW8 image data

12.1 CSI-2 configuration

The CSI-2 configuration is global to all contexts.

12.1.1 Output data rate

The MIPI_DATA_RATE register (SENSOR_SETTINGS group) controls the global parameter CSI-2 data rate. The value is set in bits per second for one lane. The exact CSI-2 data rate is adjusted by the sensor according to the EXTCLK value and the PLL capacity.

Table 95. **MIPI_DATA_RATE [31:0]**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_RATE																															

The real MIPI data rate can be slightly lower than the value programmed in register MIPI_DATA_RATE. Refer to Section 8.4: MIPI clock.

12.1.2 Data lane number

The output interface can be configured to transmitt data over 1 lane, 2 lanes, or 4 lanes.

Table 96. **LANE_NB_SEL [7:0] {0x743}**

7	6	5	4	3	2	1	0
Reserved						VALUE	

Table 97. **List of states for LANE_NB_SEL**

Value	Name	Description
0	LANES_NB_4	4 lanes
1	LANES_NB_2	2 lanes
2	LANES_NB_1	1 lane

12.1.3 Lane polarity swapping

There is individual control over the polarity of each lane (data or clock).

The OIF_LANE_PHY_SWAP register (STREAM_STATICS group) controls the polarity swapping.

Table 98. OIF_LANE_PHY_SWAP [7:0]

7	6	5	4	3	2	1	0
	Reserved		CLK	DATALANE4	DATALANE3	DATALANE2	DATALANE1

Each lane is controlled by a specific bit:

- 0 => no swapping
- 1 => swapping

12.1.4 Data lane remapping

It is possible to remap each logic lane to a physical lane.

Table 99. OIF_LANE_PHY_MAP [7:0]

7	6	5	4	3	2	1	0
DATALANE4		DATALANE3		DATALANE2		DATALANE1	

The OIF_LANE_PHY_MAP register (STREAM_STATICS group) controls lane remapping.

Table 100. OIF_LANE_PHY_MAP description

Bit field	Field name	Description Lane mapping for physical lanes 4, 3, 2, and 1
7:6	DATALANE4	0: Logic lane is mapped to physical lane 1 1: Logic lane is mapped to physical lane 2 2: Logic lane is mapped to physical lane 3 3: Logic lane is mapped to physical lane 4
5:4	DATALANE3	0: Logic lane is mapped to physical lane 1 1: Logic lane is mapped to physical lane 2 2: Logic lane is mapped to physical lane 3 3: Logic lane is mapped to physical lane 4
3:2	DATALANE2	0: Logic lane is mapped to physical lane 1 1: Logic lane is mapped to physical lane 2 2: Logic lane is mapped to physical lane 3 3: Logic lane is mapped to physical lane 4
1:0	DATALANE1	0: Logic lane is mapped to physical lane 1 1: Logic lane is mapped to physical lane 2 2: Logic lane is mapped to physical lane 3 3: Logic lane is mapped to physical lane 4

12.1.5 CSI-2 interpacket period

To match the OIF performance with the receiver performance, it is possible to adjust the minimum period between two consecutive CSI-2 packets.

This period is controlled by the OIF_INTERPACKET_DELAY register (STREAM_STATICS group).

Interpacket delay granulatiry is derived from the effective MIPI data rate. One step of delay is equal to eight times the MIPI data rate period.

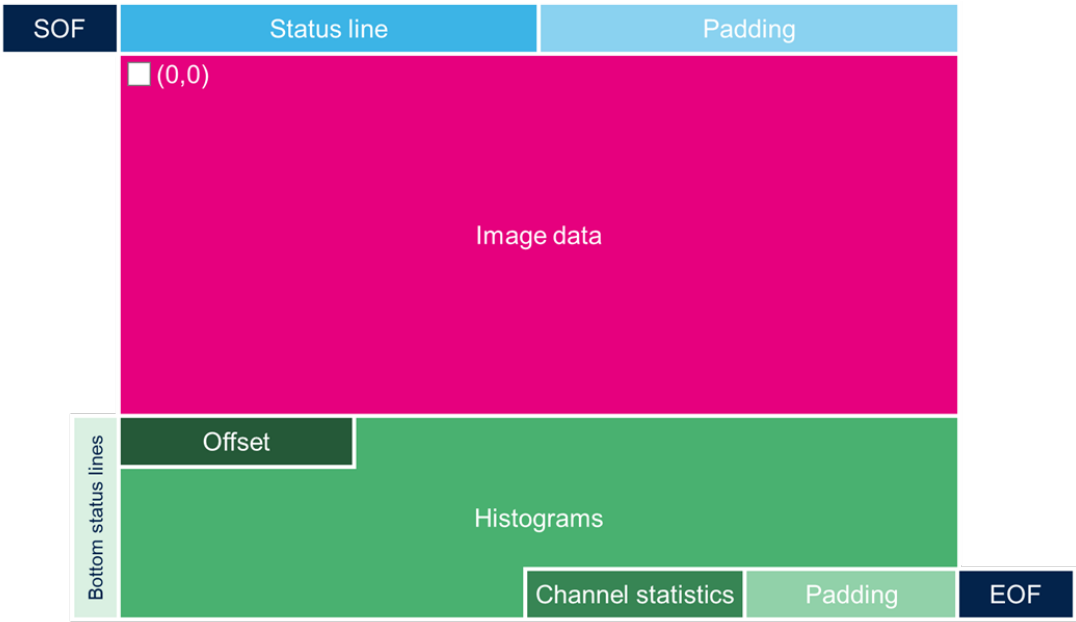
Table 101. **OIF_INTERPACKET_DELAY [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE															

12.2 Frame format

The frame is made of status lines followed by the image content.

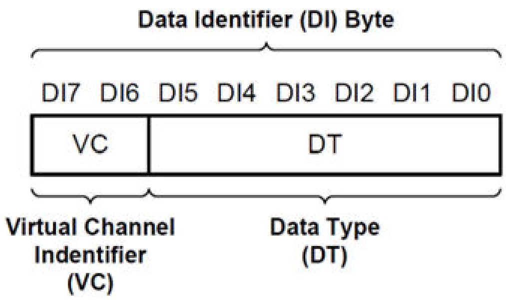
Figure 40. **Frame structure**



12.2.1 Image data type

The data type (DT) is part of a data identifier byte defined in the MIPI CSI-2 v1.1 specification.

Figure 41. **Data identifier**



The frame is composed of three parts. Each part has a different DT:

- Top status lines
- Image data
- Bottom status lines

The default DT of the image data depends on the output format (see Table 102. Default DT vs. output format).

Table 102. Default DT vs. output format

MIPI CSI-2 format	Split exposure	Single exposure
RAW8	0x33	0x2A
RAW10	0x34	0x2B
RAW12	NA	0x2C

Table 103. List of sensor configurations and corresponding default DTs

Sensor configuration	Shutter mode	VT mode	Pixel format	Split exposure	Pixel dynamic	DT
0x01	GS	Normal	P1P2P3P4	Normal	8 bits	0x2A
0x02	GS	Normal	P1P2P3P4	Normal	10 bits	0x2B
0x03	GS	Normal	P1P2P3P4	Split	8 bits	0x2A
0x04	GS	Normal	P1P2P3P4	Split	10 bits	0x2B
0x09	GS	Subsampling 2	P4	Normal	8 bits	0x33
0x0A	GS	Subsampling 2	P4	Normal	10 bits	0x34
0x0B	GS	Subsampling 4	P4	Normal	8 bits	0x33
0x0C	GS	Subsampling 4	P4	Normal	10 bits	0x34
0x0D	GS	Subsampling 32	P4	Normal	8 bits	0x33
0x0E	GS	Subsampling 32	P4	Normal	10 bits	0x34
0x1A	RS	SDR	P1P2P3P4	Normal	8 bits	0x2A
0x1B	RS	SDR	P1P2P3P4	Normal	10 bits	0x2B
0x1C	RS	SDR	P1P2P3P4	Normal	12 bits	0x2C
0x20	RS	HDR	P1P2P3P4	Normal	10 bits	0x2B
0x21	RS	HDR	P1P2P3P4	Normal	12 bits	0x2C

The DT of each ROI can be overwritten with a corresponding ROI_x_DT register (STREAM_STATICS group).

Table 104. ROI_x_DT [7:0]

7	6	5	4	3	2	1	0
Reserved		VALUE					

Note: The MIPI CSI-2 v1.1 specification defines the DT as a 6-bit coded value. The DT value should not exceed 0x3F.

12.2.2 Virtual channel

The virtual channel is part of a data identifier byte defined in the MIPI CSI-2 v1.1 specification. It is a 2-bit coded value. Each context has its own virtual channel value. The default values of the virtual channel are 0. The VIRTUAL_CHANNEL registers (STREAM_CTXx_STATIC group) configure the custom virtual channel values for each context.

Table 105. VIRTUAL_CHANNEL [7:0]

7	6	5	4	3	2	1	0
Reserved						VALUE	

The values that are allowed for the VIRTUAL_CHANNEL registers are: 0, 1, 2 and 3 (coded on 2 bits). Virtual channel values apply to all three parts of the frame: Top status lines, image data, and bottom status lines.

12.2.3 Top status lines

Top status lines, also called intelligent status line (ISL), is a CSI-2 packet transmitted before the image (frame). The size of the packet is similar to the size of an image line. The first 512 bytes are copied from the STATUS registers, then zero-padding is added.

If an image line is lower than 512 bytes, additional top status lines are created to fit the required 512 bytes of data.

The top status line is enabled by the OIF_ISL_ENABLE register (STREAM_STATICS group). The default value is enabled.

Table 106. OIF_ISL_ENABLE [7:0]

7	6	5	4	3	2	1	0
Reserved						VALUE	Reserved

Table 107. List of states for OIF_ISL_ENABLE

Value	Output ISL
0	Disable
1	Enable

Table 108. OIF_DT_ISL [7:0]

7	6	5	4	3	2	1	0
Reserved		OIF_DT_ISL					

The default data type OIF_DT_ISL for the top status line is 0x2C as the default image configuration.

Any other valid data type such as 0x12 (Embedded 8-bit non image data) or generic long packet data types can be programmed instead. When 0x00 is programmed in OIF_DT_ISL, the sensor transmits the top status line with 0x12 data type value.

12.2.4 Image data

The output data are compliant with MIPI CSI-2 specifications v1.1. The figures in the subsections below are extracted from the official MIPI specifications.

12.2.4.1 RAW8 data format

Figure 42. **RAW8 transmission**

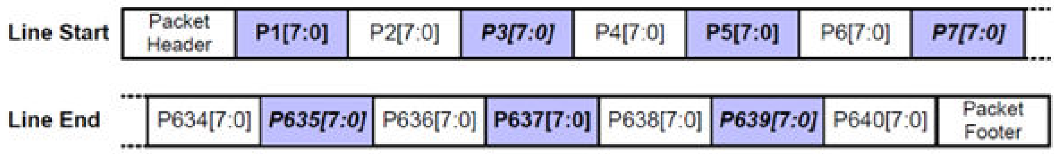
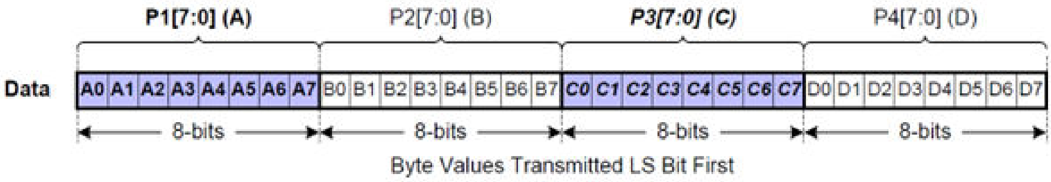


Figure 43. **RAW8 data transmission on the CSI-2 bus**



12.2.4.2 RAW10 data format

Figure 44. **RAW10 transmission**

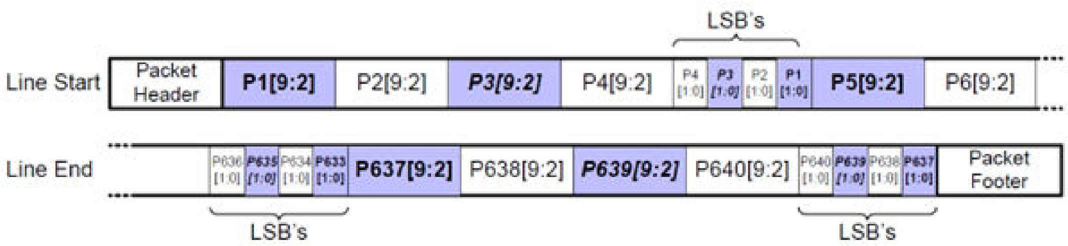
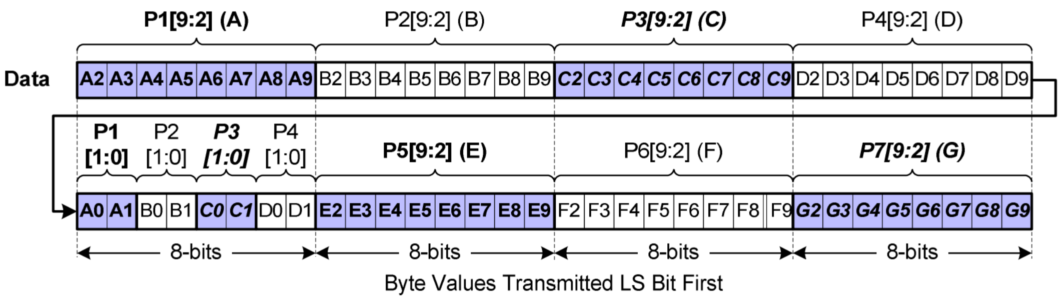


Figure 45. **RAW10 data transmission on the CSI-2 bus**



12.2.4.3 *RAW12 data format*

Figure 46. **RAW12 transmission**

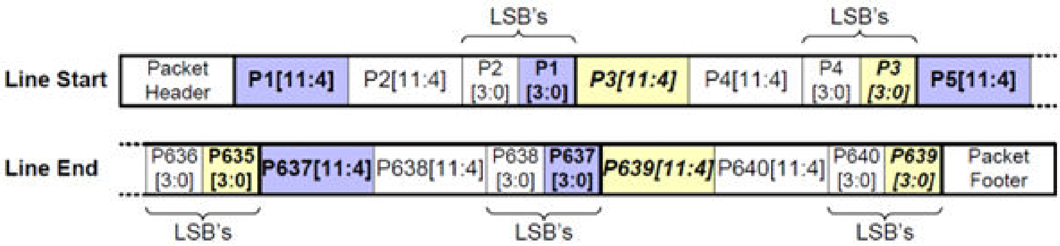
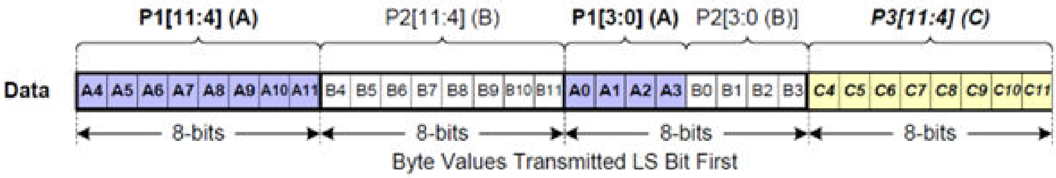


Figure 47. **RAW12 data transmission on the CSI-2 bus**



12.2.5 Bottom status lines

The general organization of the bottom status lines is as follows:

- Offset (256 bytes)
- Histogram (if transmitted)
  - 2x2048 or 3x2048 elements (4 bytes each)
- Channel statistics
  - 3x16 accumulators (4 bytes each)
  - 3x pixel counters (4 bytes each)

The bottom status lines are transmitted at the end of each frame. It is always an even number of lines. Each data packet is aligned on an image line. Disabling of the bottom status lines is controlled by the BOTTOM_STATUS_LINE_DISABLE register (DEBUG group).

Table 109. **BOTTOM_STATUS_LINE_DISABLE [7:0]**

7	6	5	4	3	2	1	0
Reserved							VALUE

Table 110. **List of states for BOTTOM_STATUS_LINE_DISABLE**

Value	Output bottom status lines
0	Enable
1	Disable

Table 111. **OIF_DT_STAT [7:0]**

7	6	5	4	3	2	1	0
Reserved		OIF_DT_STAT					

The default data type OIF_DT_STAT for the bottom status line is 0x2C as the default image configuration. Any other valid data type such as 0x12 (Embedded 8-bit non image data) or generic long packet data types can be programmed instead. The value 0x00 is not supported.

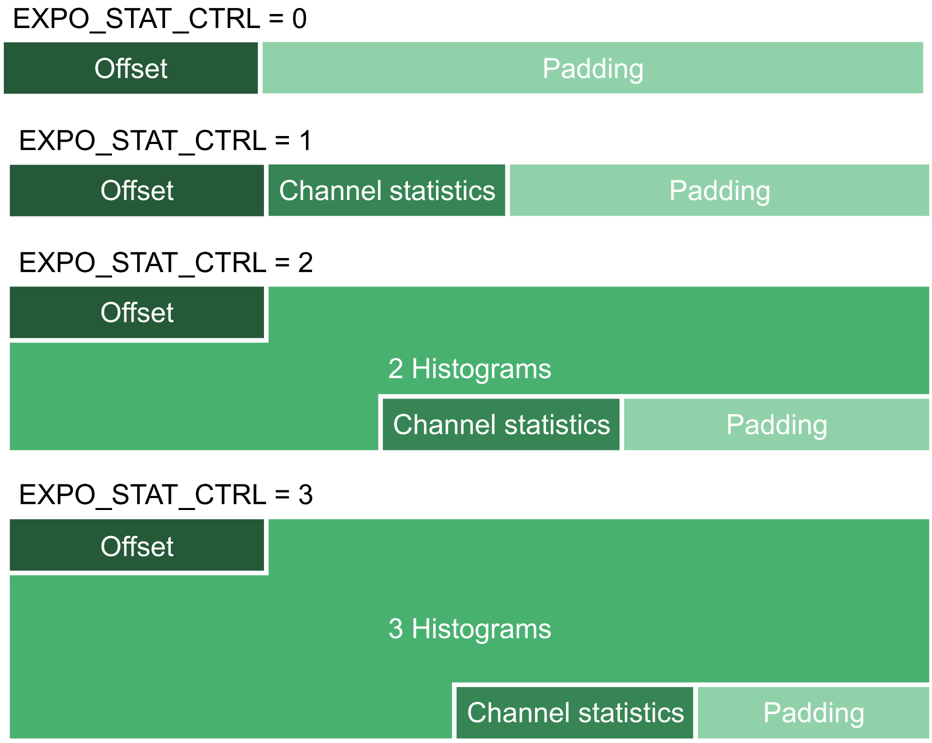


Figure 48. Organization of the bottom status lines

The organization of the bottom status lines depends on the EXPO_STAT_CTRL register value (see Section 10.9: Exposure statistics). The histogram configuration is as described in Section 10.9.3: Histograms. Channel statistics configuration is described in Section 10.9.2: Channel statistics.

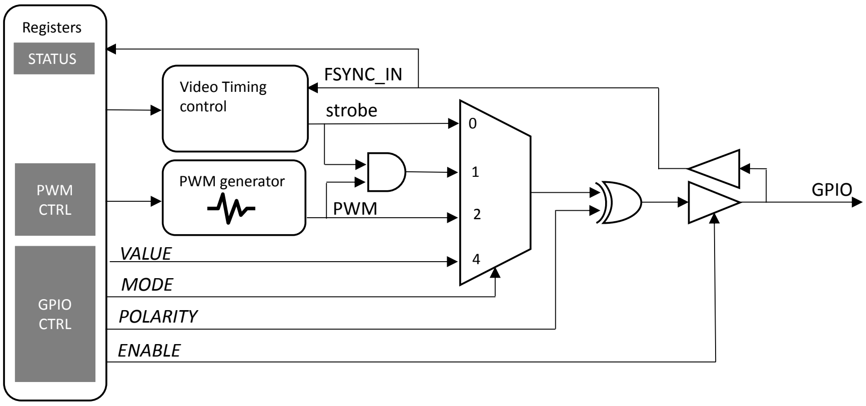
13 GPIOs

The VB5943 and VD5943 sensors have four configurable GPIOs. They are configured in SW STANDBY state but they are active in STREAMING state only.

13.1 Configuration

GPIO configurations are global to all contexts, where a mode is assigned to a GPIO. However, GPIO active/ inactive control mode is specific to each context.

Figure 49. Overview of GPIO configuration



13.1.1 GPIO global configuration

Each GPIO configuration is controlled by its specific GPIO_x_CTRL register of the STREAM_STATICS group. The four registers (for the four GPIOs) must be updated in SW STANDBY state.

Table 112. GPIO_x_CTRL [7:0]

7	6	5	4	3	2	1	0
Reserved		POLARITY	VALUE	MODE			

Table 113. GPIO_x_CTRL description

Mode value	Mode name	Description
0	Strobe	The GPIO is asserted during the integration of the P4 pixels. Refer to the section below for the configuration of the rising and falling edges.
1	PWM_strobe	The GPIO sends a pulse width modulation during the integration of the P4 pixels.
2	PWM	The GPIO sends a pulse width modulation. Refer to the section below for the configuration of PWM generation.
3	GP_IN	The pad is configured in input. The value is sampled at every frame. The logic value (0 or 1) is available in the GPIO_STATUS register.
4	GP_OUT	The pad is configured in output. The VALUE field controls the level.
5	FSYNC_IN	External synchronization for the follower mode (rising edge only)
7	FSYNC_OUT	Video timing output pulse for new start of frame (only for GPIO_0)

- The POLARITY field controls a logic inverter. The GPIO output is inverted.
- The VALUE field is used in GP_OUT mode to set the output.

13.1.2 GPIO individual control

The four GPIOs are individually controlled by the GPIO_CTRL register of each context (STREAM_CTXx_STATIC group).

When a GPIO is enabled, the output is driven by the source selected by the MODE field.

When a GPIO is disabled, the output is forced at low level (weak pull-down active).

The four registers must be updated in SW_STANDBY state.

Table 114. GPIO_CTRL [7:0]

7	6	5	4	3	2	1	0
Reserved				GPIO3	GPIO2	GPIO1	GPIO0

Table 115. GPIO_CTRL description

GPIOx	Description
0	Disable
1	Enable

13.2 Strobe mode

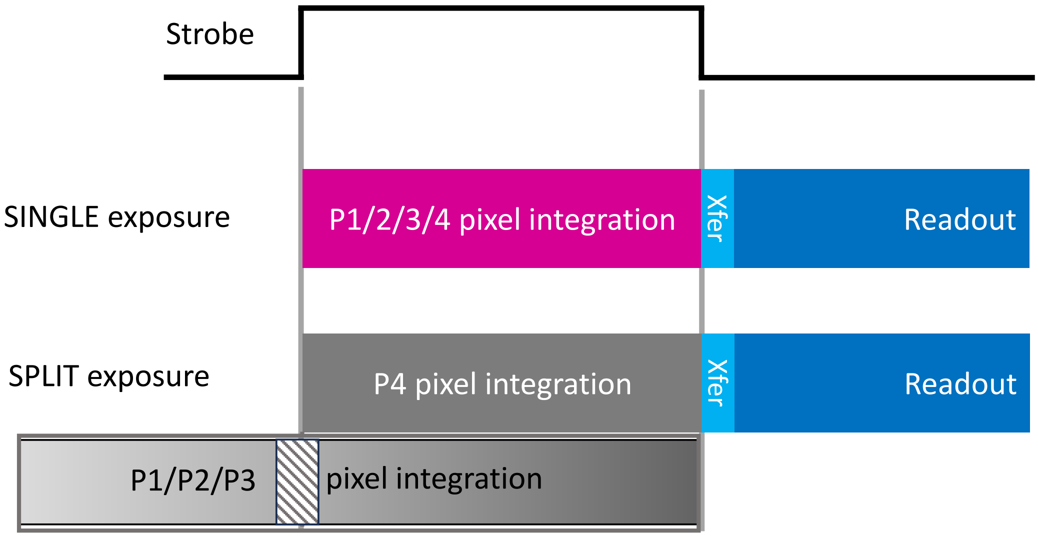
In strobe mode, the video timing controller drives the GPIO. The signal is synchronized with the integration period of the P4 pixels. This mode is active in global shutter only. It can be used to drive an external IR illuminator.

When the sensor's configuration uses the split exposure mode, two concurrent exposures INTEGRATION_TIME_P4 and INTEGRATION_TIME_PRIMARY take place.

The sensor manages the integration sequence in order to have INTEGRATION_TIME_P4 and INTEGRATION_TIME_PRIMARY ending at the same time.

The STROBE signal, which is managed by INTEGRATION_TIME_P4, partially or totally covers the INTEGRATION_TIME_PRIMARY.

Figure 50. Strobe signal without delay



An illuminator system has a turn-on time and a turn-off time. To adjust the signal to the illuminator settings, use the STROBE_START_DELAY and STROBE_END_DELAY registers. They control the rising time and the falling time of the strobe with regard to the integration period. These registers are 8-bit signed, and are expressed in line time. They are global to all contexts and belong to the STREAM_STATICS group. The value of STROBE_END_DELAY must be set between -128 and 0, inclusive.

Table 116. **STROBE_START_DELAY [7:0]/STROBE_END_DELAY [7:0]**

7	6	5	4	3	2	1	0
DELAY							

These registers control respectively the shift on the rising edge and falling edge of the strobe.

Figure 51. **Strobe with delay (example 1)**

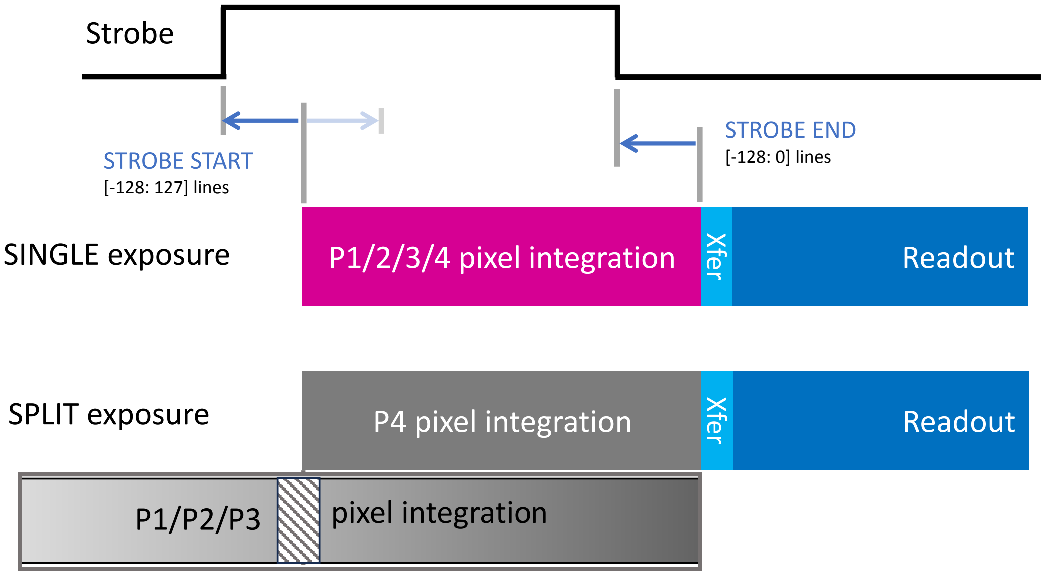
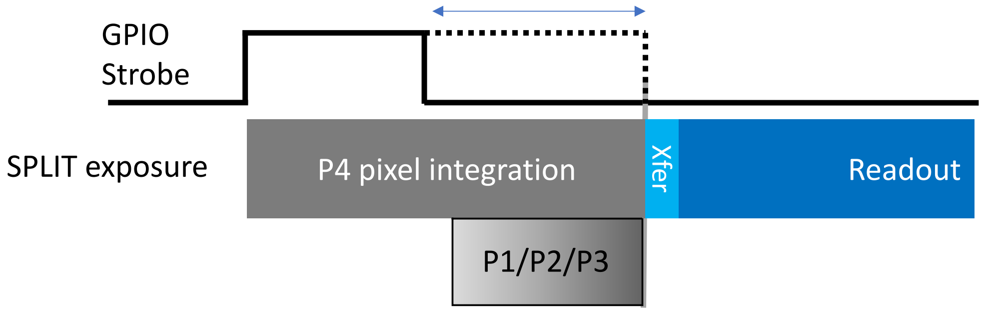


Figure 52. **Strobe with delay (example 2)**



13.3 PWM mode

In this mode, the sensors generate a modulated signal. The parameters of the modulation are controlled by the PWM_CTRL register of the STREAM_STATICS group.

- When PWM mode is selected, the signal is continuously emitted during streaming.
- When PWM_STROBE mode is selected, the signal is emitted during the integration of the P4 pixels only.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKDIVISOR																Reserved												DUTYCYCLE			

- PWM frequency is computed from the CLKDIVISOR field assuming an MCU clock at 150 MHz.

$$PWM_{frequency} = \frac{MCU\_clock}{ClkDivisor + 1}$$

- Low period duty cycle is computed from the DUTYCYCLE field

$$Duty_{Cycle} = \frac{16 - DutyCycle}{16}$$

13.4 GP_IN mode

In GP_IN mode, the logical value of the GPIO is sampled every frame when the video stream is enabled (STREAMING state). The value is available in the GPIO_STATUS register of the STATUS group.

The register is not updated in SW STANDBY state.

Table 117. GPIO_STATUS [7:0]

7	6	5	4	3	2	1	0
Reserved				GPIO_3	GPIO_2	GPIO_1	GPIO_0

13.5 GP_OUT mode

In GP_OUT mode, the output of the GPIO is controlled by the VALUE and POLARITY field.

Refer to Section 13.1.2: GPIO individual control.

13.6 FSYNC_IN

In FSYNC_IN mode, the GPIO is used in follower synchronization mode to start the integration of each frame. Each pulse on the GPIO is converted into a single frame. The triggering event is rising edge only.

It is possible to delay the start of the exposure with regards to the FSYNC_IN rising edge. The global register FSYNC_IN_DELAY (STREAM_STATICS group) controls the delay expressed in line time. The delay is in the range 0 to 512. The default value is 0.

Table 118. FSYNC_IN_DELAY [15:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						VALUE									

When FSYNC_IN_DELAY is set to 0, the latency from FSYNC_IN triggering edge up to the start of integration is in the range of 12 to 14 lines.

13.7 FSYNC_OUT

In FSYNC_OUT mode, the GPIO is used to emit a short synchronization pulse at the start of each new frame.

The width of this pulse is equal to 20 pixel clock cycles (214 ns when the pixel clock frequency is at 93.5 MHz).

This pulse is generated at least six lines prior to the start of exposure.

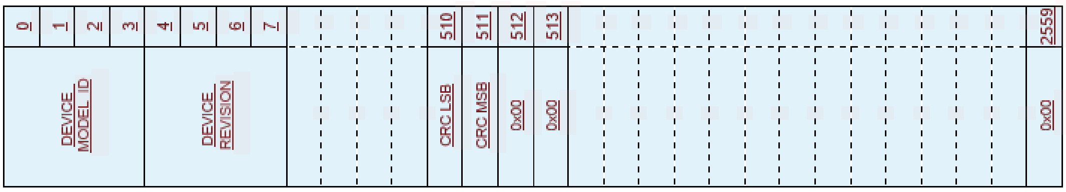
Only GPIO_0 supports this feature.

14 Miscellaneous

14.1 Top status lines

The content of the top status lines is protected by a 16-bit CRC (through the CCITT). The first 510 bytes are relevant (see Figure 53). The CRC is available at offset 510 (LSB) and offset 511 (MSB).

Figure 53. **Content of the top status lines**



14.2 Temperature

The devices embed two thermal sensors. Temperature is read at each frame in the STREAMING state. Alternatively, if a THSENS_READ command is sent, temperature is read when in the SW STANDBY state.

Results are available in the TEMPERATURE_1 and TEMPERATURE_2 status registers as 16-bit signed integers.

Table 119. **TEMPERATURE_1/_2 [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH_DEGREE															

14.3 System state and error codes

If a command cannot be executed or an error occurs, the device enters halt state. At this point, the SYSTEM_FSM_STATE register is 0x0006, and the device must be rebooted.

When the sensor is in halt state, the SYSTEM_ERROR register is different from 0x0000.

Table 120. **SYSTEM_ERROR [15:0]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSTEM_ERROR															

The list of error codes, and their corresponding descriptions are provided in the reference manual (RM0532).

Revision history

Table 121. **Document revision history**

Date	Version	Changes
20-Mar-2026	1	Initial release

Contents