General description

The epc670 is a fully integrated 3D-cwTOF imager with a resolution of 320 × 240 pixels (QVGA). As a system on chip, the epc670 contains next to the CCD pixel field the complete control logic to operate the device. The output of the chip is 12 bit DCS distance data per pixel, which are accessible through a mobile industry processor interface (MIPI).

Only a few additional components are needed to build a complete 3D camera. A distance resolution in the millimeter range for distances up to 20 meters while ensuring eye safety standards is feasible. Up to 700 full frame TOF images per second are delivered in rolling mode. The extremely high sensitivity of the chip allows low illumination power and reduced overall power consumption compared to other TOF imagers.

Key features

- Exceptional QE in NIR: 72% @ 905 nm
- High ambient light immunity for outdoor applications
- Ultra-high frame rate: up to 700 TOF fps (54 Mpixel/s)
- High sensitivity of 8 e-/LSB
- High TOF dynamic range: 32 dB for a single integration time
- Simultaneous grayscale image, day and night
- Mobile industry processor interface (MIPI, CSI-2 over D-PHY)
- Internal or external modulation clock for multi-camera setups
- SIL2 ready

Block diagram

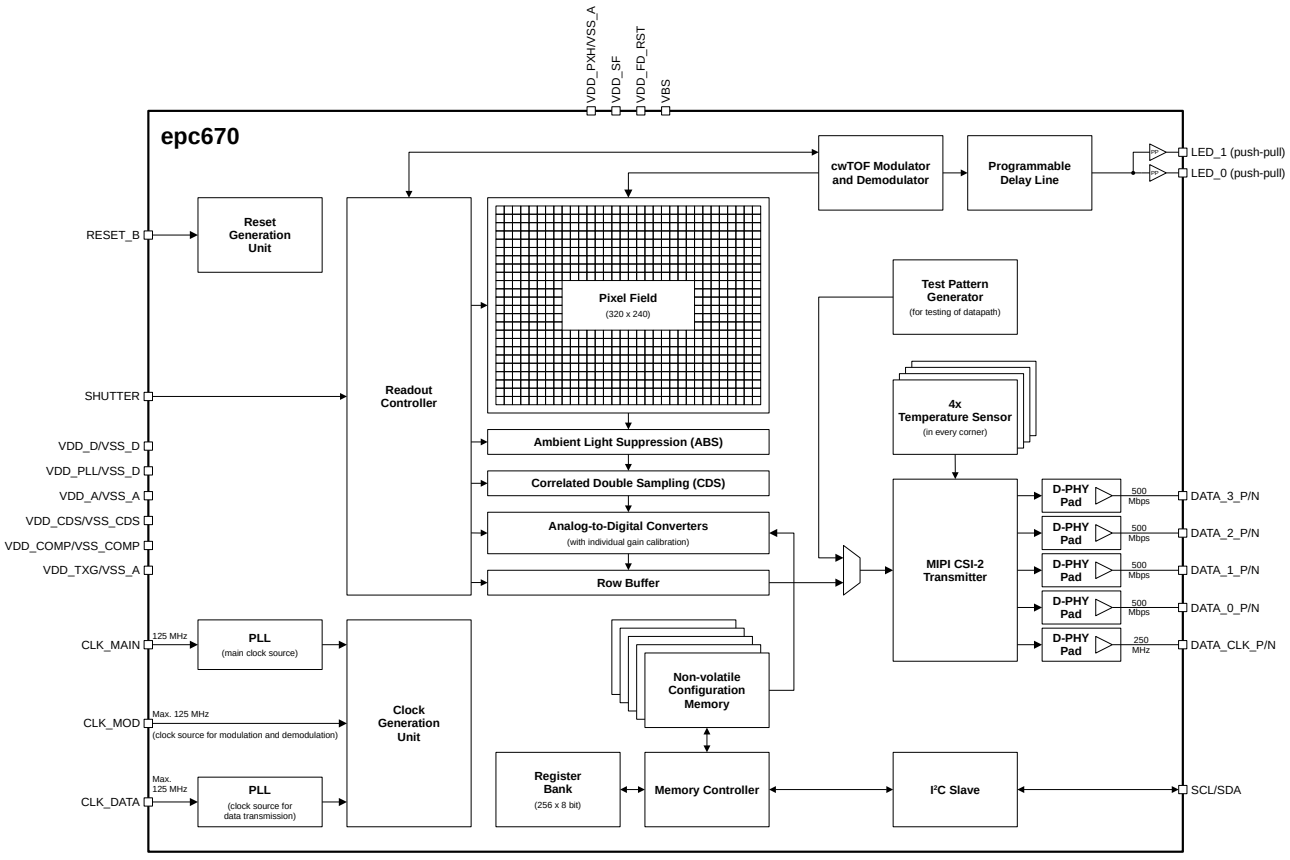

Figure 1: High-level functional block diagram

Main features

- **General**
  - Full monolithic design
  - 320 × 240 pixel field
  - Global shutter
  - Up to 54 million 3D data points per second
  - Back-illuminated, 100% fill factor pixel design
  - Active illumination range: visible to IR up to 1000 nm
    - Anti-reflective coating optimized for 900 to 940 nm
  - Automatic suppression of ambient light on pixel level in the charge domain
  - Full daylight operation
  - Selectable gain in the range of 0.25 to 4
  - 4 integrated temperature sensors
- **Optical data**
  - Pixel pitch of 20 µm × 20 µm
  - Sensitivity of 8 electrons per LSB (@ gain of 4)
  - 12 bit ADC resolution
  - TOF dynamic range of >32 dB (one integration time)
- **Mobile industry processor interface (MIPI)**
  - D-PHY v1.2 physical layer
    - 4 differential data lanes (500 Mbps per lane, 2 Gbps in total)
    - 1 differential clock lane (250 MHz)
  - CSI-2 v2.1 protocol layer
- **I²C control interface**
- **Integrated non-volatile memory**
  - Calibration
  - User-programmable parameters
  - Unique chip ID
- **Frame configuration table**
  - Pre-configuration of up to 16 DCS frames per burst triggered by a single shutter (e.g., integration time, modulation frequency, region of interest, etc.)
- **Flexible clocking**
  - Separate input clocks for main part of the sensor, modulation and data transmission
  - Up to 125 MHz input clock each
- **Functional safety features**
  - Predefined test patterns
    - Full scale (stripes)
    - Ramp (gradient)
    - PN9 (pseudo-random)
  - Metadata including temperature data per MIPI frame
  - CRC protection of transmitted data on MIPI CSI-2 protocol level
  - Selectable ADC output test values which can be used to check ADC counters, data handling and data transmission
  - Parity checking for processed readout sequencer code
  - Register and sequencer RAM dump over MIPI
- **Packaging**
  - Back-illuminated flip-chip SMD mounting
  - Cost optimized 79 pin CSP, 11.7 mm × 9.3 mm
  - RoHS compliant
- **Other data**
  - -40 °C to +125 °C operating range

Measurement modes

- **Modulation**
  - Sinusoidal modulation
  - Selectable modulation frequencies from 0.625 MHz to 31.25 MHz resulting in unambiguity distances of 4.8 m to 240 m
  - 2 illumination control pins which can be controlled individually
- **Distance measurement modes**
  - 700 fps with 4 DCS rolling mode (full pixel field)
  - 350 fps with 2 DCS (full pixel field)
  - 175 fps with 4 DCS (full pixel field)
  - Ultra-fast measurement by reduction of the image field (ROI)
  - Shutter release input for precise start/stop and single/continuous measurement control
- **Non-distance light measurement modes**
  - Ambient light measurement (grayscale image without active illumination)
  - Grayscale image with active illumination
- **Resolution reduction**
  - Readout of only every second row and/or column
  - Increased frame rate for reduced number of rows
- **Region of interest (ROI)**
  - Rectangular sub-pixel-field readout
  - Increased frame rate or reduced power consumption

Table of contents

1. Typical application setup

A typical system-level setup is shown in Figure 2. After ramping up the supplies, enabling the clock sources and releasing the reset, the epc670 can be configured over I^2C, e.g. to set the integration time. ToF DCS image bursts can then be obtained by pulsing the shutter pin. Alternatively, DCS bursts can be triggered by an I^2C command. This automatically generates the modulated LED control signal for the integration phase of every image of a burst. The image data is transmitted over a MIPI D-PHY interface consisting of four differential data lanes and one differential clock lane. A MIPI CSI-2 receiver is required on the controller side to extract the image data from the stream. Based on the received data, distance and amplitude can then be computed for every pixel.

A separate safety illumination unit can be used to check the sensor operation based on optical input.

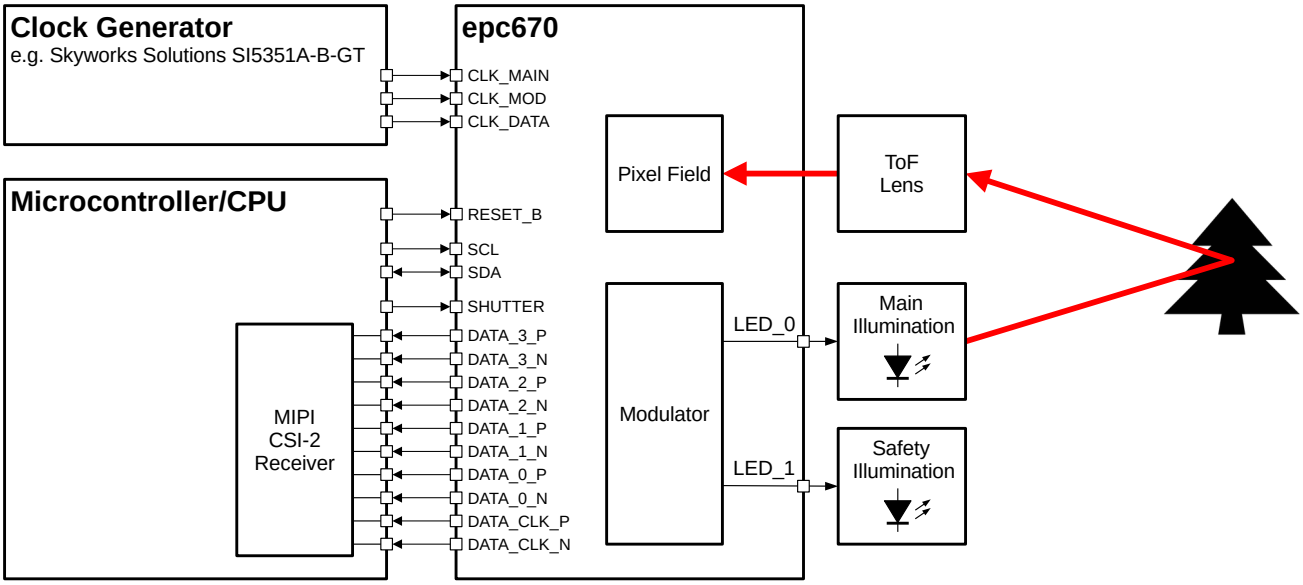

Figure 2: Typical application diagram

2. Characteristics

All characteristics are at typical operational ratings, T_A = +25 °C, modulation frequency 10 MHz, supply voltages at nominal value, unless otherwise stated.

2.1. Optical characteristics

Table 1: Imager characteristics

Parameter	Description	Conditions/Comments	Value	Unit
Res	Resolution	Active pixels (default)	320 × 240	pixel
A_{PIXEL}	Pixel photosensitive area	100% fill factor	20 × 20	μm^2
A_{SENSOR}	Pixel field area	Area of sensor array	6.4 × 4.8	mm^2

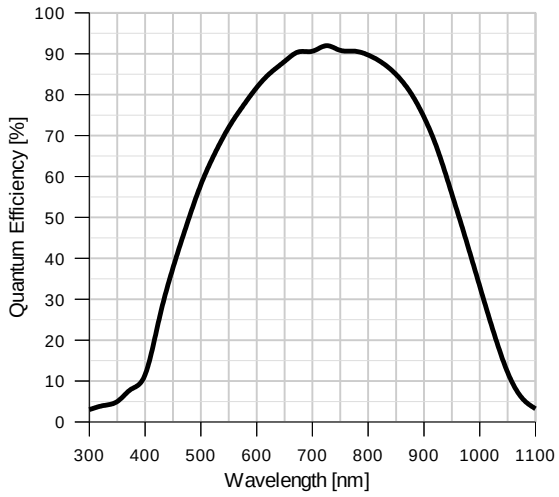

Figure 3: Typical quantum efficiency

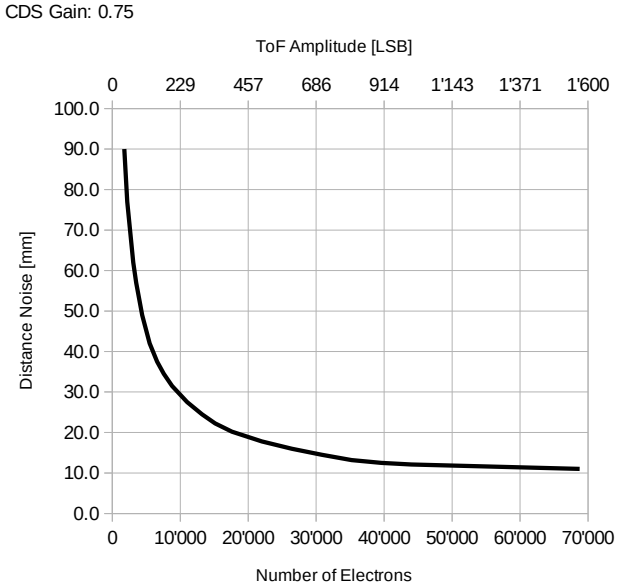

Figure 4: Distance noise vs. signal level, single shot, 4 DCS, no ambient light

2.2. Imaging characteristics

Table 2: Imaging characteristics

Parameter	Description	Min.	Typ.	Max.	Unit
TOF sensitivity S_{TOF}	• Modulation frequency: 10 MHz • ToF amplitude: 1'500 LSB • CDS gain: 4 • Wave length: 940 nm	250	290	320	$\frac{pW/mm^2}{LSB}$
ToF amplitude PRNU	ToF amplitude photo response non-uniformity, @ 1000 LSB		4%		
DRNU	Distance response non-uniformity, @ 1'500 LSB, no compensation	50 (1)	250 (2)		mm
Distance noise (dark)	Temporal distance noise (1σ) without ambient light • ToF amplitude: 500 LSB (~21.5k e-) @ CDS gain 0.75	10 (1)	18	25	mm
I_{Dark}	Dark current @ CDS gain 4 (grayscale signal level increase during integration in the dark)		15	30	LSB/ms
SE	Shutter efficiency @ 940 nm wave length	98.0%	99.0%	99.5%	
PLS	Parasitic light sensitivity @ 940 nm wave length	0.5%	1.0%	2.0%	
S_{BW}	Grayscale sensitivity • 100 µs integration time • CDS gain: 4	80	90	100	$\frac{pW/mm^2}{LSB}$
DSNU	Dark signal non-uniformity @ CDS gain 0.75	15	20	30	LSB
PRNU	Photo response non-uniformity @ CDS gain 0.75		5%		

Notes:

1 Target for typical case

2 Diagonal distance shift across the pixel field

2.3. Ambient light suppression (ABS)

An important function of the 3D TOF pixel is the ambient light suppression. It removes DC or low frequency modulated light caused by sunlight, room illumination, etc. from the modulated light generated by the camera illumination. The amount of collected ambient light is proportional to the integration time. The longer the integration time, the more unwanted light will be collected. It is good practice to keep the integration time for TOF imaging below 1 ms. In addition, the use of optical band-pass filters to block the unwanted spectrum of light is mandatory for outdoor applications.

Table 3: Ambient light suppression

Parameter	Description	Integration time	Wavelength	Typical value	Unit
$E_{e,max}$	Max. irradiance, DC light	100 µs	940 nm	60	µW/mm²
		800 µs	940 nm	7.5	µW/mm²
	Full well capacity of storage gates	n/a	n/a	5	Me^-
	Ambient light energy density at imager surface corresponding to full well capacity of storage gates	n/a	940 nm	6	nJ/mm²

2.4. Operating conditions and electrical characteristics

Table 4: Operating conditions and electrical characteristics

Parameter	Description	Conditions/Comments	Min.	Typ.	Max.	Unit
V_{DD_D}, V_{DD_PLL}	Digital supply voltage	Ripple [1] < ± 20 mV	1.71	1.80	1.98	V
$V_{DD\_A/CDS/COMP/TXG}$	Analog supply voltage [2]	Ripple [1] < ± 20 mV	4.9	5.0	5.1	V
V_{DD_PXH}	High voltage pixel driver supply voltage [2]	Ripple [1] < ± 20 mV	9.5	10	10.5	V
V_{DD_SF}	Source follower supply voltage	Noise shall be minimum.	9.5	10	10.5	V
$V_{DD\_FD\_RST}$	Floating diffusion reset supply voltage	Noise shall be minimum.	8.5	9	10	V
V_{BS}	Backside supply voltage		-20.0	-15.0	-14.5	V
I_{VDD_D}	Total digital supply current, including MIPI D-PHY supply currents		45	230	250	mA
I_{VDD_PLL}	PLL supply current	For both PLLs together	6	7	8	mA
$I_{VDD\_A/CDS/COMP/TXG}$	Analog supply current		80	225	250	mA
I_{VDD_PXH}	High voltage pixel driver supply current		0.25	0.30	0.35	mA
I_{VDD_SF}	Source follower supply current		11.0	12.5	14.0	mA
$I_{VDD\_FD\_RST}$	Floating diffusion reset supply current		< 1	< 1	< 1	mA
I_{VBS}	Backside supply current [3]		-20	-4.5	-2.9	mA
$I_{LED\_0/1}$	LED_0/1 output sink/source current (DC)				5	mA
R_{PD_reset}	Internal pull-down resistor for RESET_B			590		kΩ
R_{PU_gpio}	Internal pull-up resistor for GPIOs			160		kΩ
R_{PD_gpio}	Internal pull-down resistor for GPIOs			160		kΩ

Notes:

1 Min. and max. voltage values include noise and ripple voltages.

2 Voltage supplies have direct influence on measurement performance. They must be properly decoupled.

3 The current is increasing for more ambient light and it is further increasing the more negative the applied V_{BS} is.

2.5. Power consumption

The epc670 has several power states/levels during the different operation phases which are described in Table 5 and Figure 5.

Table 5: Typical average power consumption levels at different operating states

Power state	Power [mW]	Operation description
RESET	95	All supplies are powered-up and stable, RESET_B = 0, PLLs and all system clocks are not running
IDLE (STATIC)	745	RESET_B = 1, PLLs and all system clocks are running, waiting for shutter
INTEGRATION	1425	Image acquisition (active illumination and demodulation in pixel)
READOUT	1760	Integration finished, readout of pixel field, AD conversion and data transmission

Note: All the values in Table 5 are assuming a backside supply current I_{VBS} of 5 mA resulting in a contribution of 75 mW.

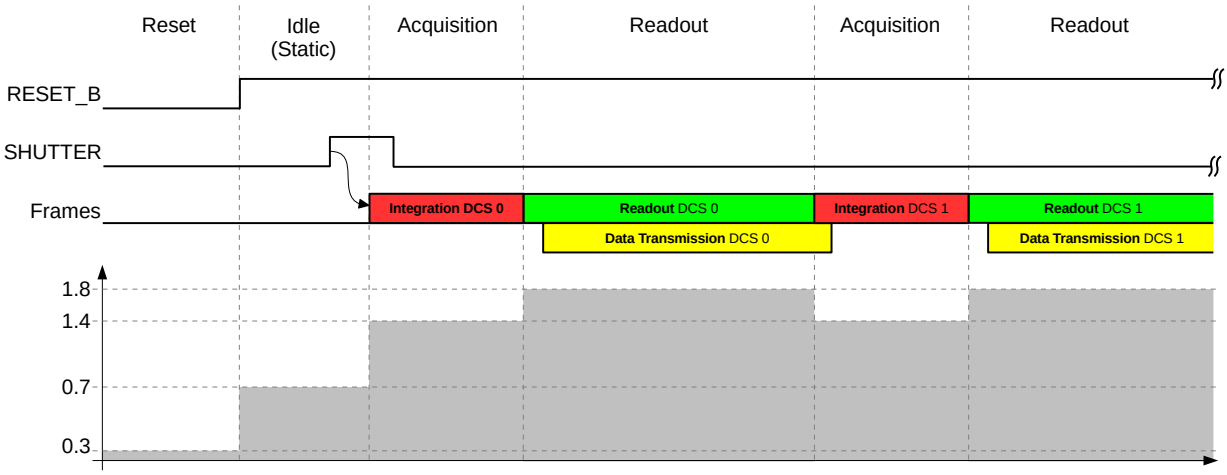

Figure 5: Power consumption levels for different operating states

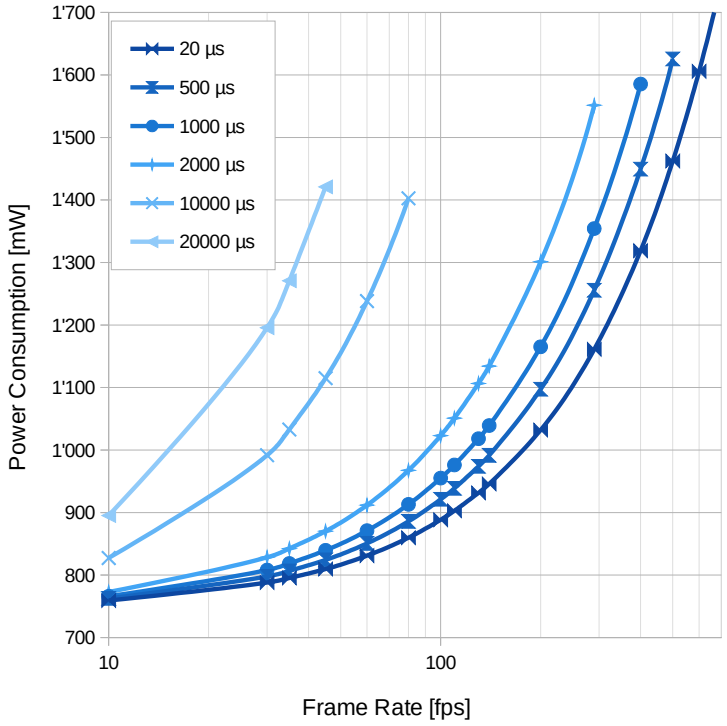

Figure 6: Typical average power consumption for static power control depending on DCS frame rate (usually 4 DCSs per distance image)

2.6. Absolute maximum ratings

Table 6: Absolute maximum ratings

Parameter	Conditions
Supply voltage V_{DD_D}, V_{DD_PLL}	-0.5 V ... +2.0 V
Supply voltage V_{DD_A}, V_{DD_CDS}, V_{DD_COMP}, V_{DD_TXG}	-0.5 V ... +5.5 V
Supply voltage V_{DD_PXH}	-0.5 V ... +13.5 V
Supply voltage V_{BS}	-20.5 V ... +0.5 V
Voltage to any pin in the same V_{SC} supply class.	$V_{SC,min}$ - 0.3 V … $V_{SC,max}$ + 0.3 V
ESD rating	JEDEC HBM class 1C (1 kV to < 2 kV)
Junction temperature (T_J)	-40 °C to +125 °C
Relative humidity	0% ... 95%, non-condensing

2.7. Timing parameters

Table 7: Timing parameters

Parameter	Description	Conditions	Min.	Typ.	Max.	Unit
f_{CLK_MAIN}	Main clock input frequency			125	125	MHz
$t_{CLK\_MAIN\_jitter}$	Main clock input phase jitter	Peak-to-peak, cycle to cycle			50	ps
f_{CLK_DATA}	Data clock input frequency			125	125	MHz
$t_{CLK\_DATA\_jitter}$	Data clock input phase jitter	Peak-to-peak, cycle to cycle			50	ps
t_{reset}	Time for main chip reset (ad pad)	Supplies ramped up and stable	100			ns
$t_{startup}$	Start-up time			721		µs
t_{PLL}	PLL lock time				700	µs
t_{ADC_cal}	ADC gain calibration time	Following $t_{startup}$ in Normal Mode		342		µs
f_{CLK_MOD}	Modulation clock frequency (at pad)	LED modulation frequency = f_{CLK_MOD} / 4	0.4	40	125	MHz
f_{LED}	LED modulation frequency		0.1	10	31.25	MHz
$D_{unambiguity}$	Unambiguity distance		4.8	15	1'500	m
t_{DL_coarse}	Delay line resolution coarse	49 steps		2.1		ns
t_{DL_max}	Max. delay of the delay line				105	ns
d_{DL_max}	Max. virtual distance with the DL	$d_{DL\_max} = c \times t_{DL\_max} / 2$			15.75	m
f_{SCL}	I^2C data rate			400	1'000	kbit/s
t_{EEPROM_Write}	Time for EEPROM write access	Waiting time per byte			25	ms

2.8. Frame rate

Figure 7 illustrates the achievable DCS frame rate depending on the selected integration time. For short integration times, mainly the time required to read out the pixel field determines the achievable frame rate.

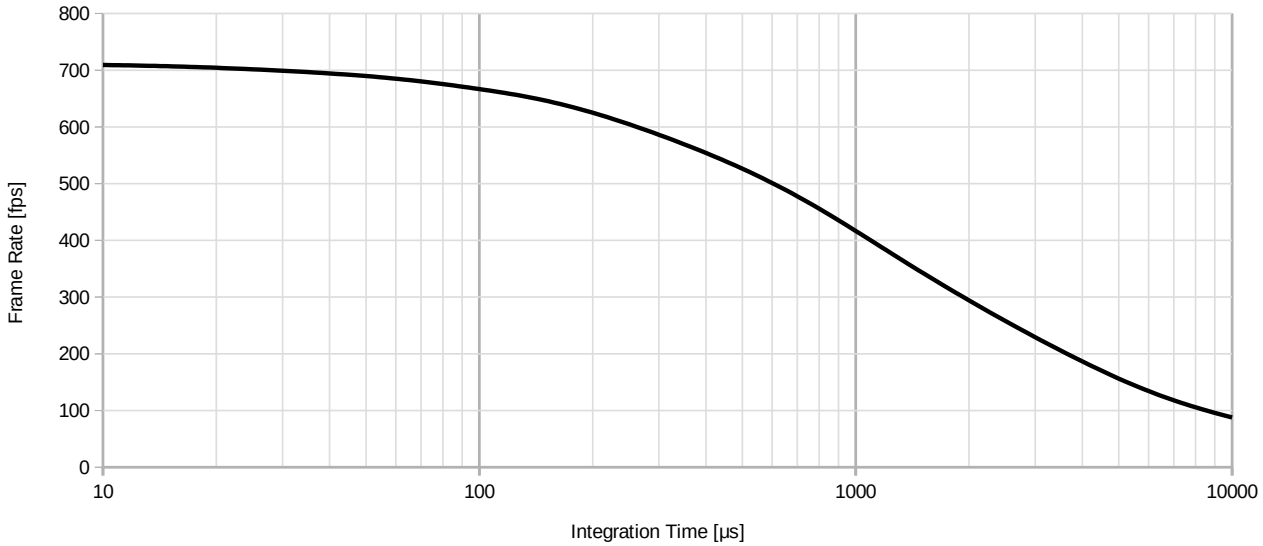

Figure 7: Achievable DCS frame rate based on integration time

Furthermore, Table 8 lists the system level timing parameters defining the frame rate based on default settings (incl. sequencer program default_3).

Table 8: System level timing parameters defining the frame rate

Parameter	Description	Value	Unit
t_{ADC}	Time for an AD conversion with an ADC counter clock frequency of 250 MHz **Note:** - Four rows are converted in parallel. - Two conversions are required per row (even and odd columns). - Therefore, four rows have to be transmitted after two AD conversions.	10.4	µs
t_{ADC_frame}	Time required for AD conversion of a whole frame (320 × 240 pixels, 120+1 conversion cycles in total) with an ADC counter clock frequency of 250 MHz	1.26	ms
t_{TX_row}	Time to transmit a row (including some overhead) over MIPI: - clock frequency of 250 MHz (500 Mbps per data lane) - 4 data lanes	3.3	µs
t_{TX_frame}	Time to transmit a frame (including some overhead) over MIPI: - clock frequency of 250 MHz (500 Mbps per data lane) - 4 data lanes	0.8	ms
t_{integ_typ}	Typical integration time	0.1	ms
t_{tof_frame}	Time for a distance measurement (acquisition and data transmission) based on 4 DCS frames **Note:** Since data transmission is faster than AD conversion, the latter is dominant.	5.5	ms
f_{tof_frame}	3D TOF frame rate based on 4 DCSs (100 µs integration time)	182	fps

2.9. Distance measurement temperature drift

The contributions of temperature effects on distance measurements are characterized in Table 9.

Table 9: Optical characteristics (10 MHz modulation frequency)

Parameter	Description	Conditions / Comments	Min.	Typ.	Max.	Unit
TC_{PIX}	Pixel	Valid up to 105 °C		20		mm/K
TC_{DL}	Delay line	Per coarse step		0.65		mm/K

2.10. Other characteristics

Table 10: Other characteristics

Parameter	Description	Conditions / Comments	Min.	Typ.	Max.	Unit
P_{Pk}	Power dissipation (average)	30 ToF fps, 1 ms integration time See Chapter 2.5 for more details.		1000		mW
R_{Th}	Thermal resistance	With solder balls and underfill, excl. PCB			1	K/W
T_{OP}	Operating temperature chip/circuitry		-40	60	125	°C
$T_{OP, optical}$	Optical performance temperature		-40	60	105	°C
Reliability	AEC-Q100 (upon request)	Grade 2	-40		125	°C
Environment	RoHS compliance		compliant			

3. Pinout

3.1. Pin mapping

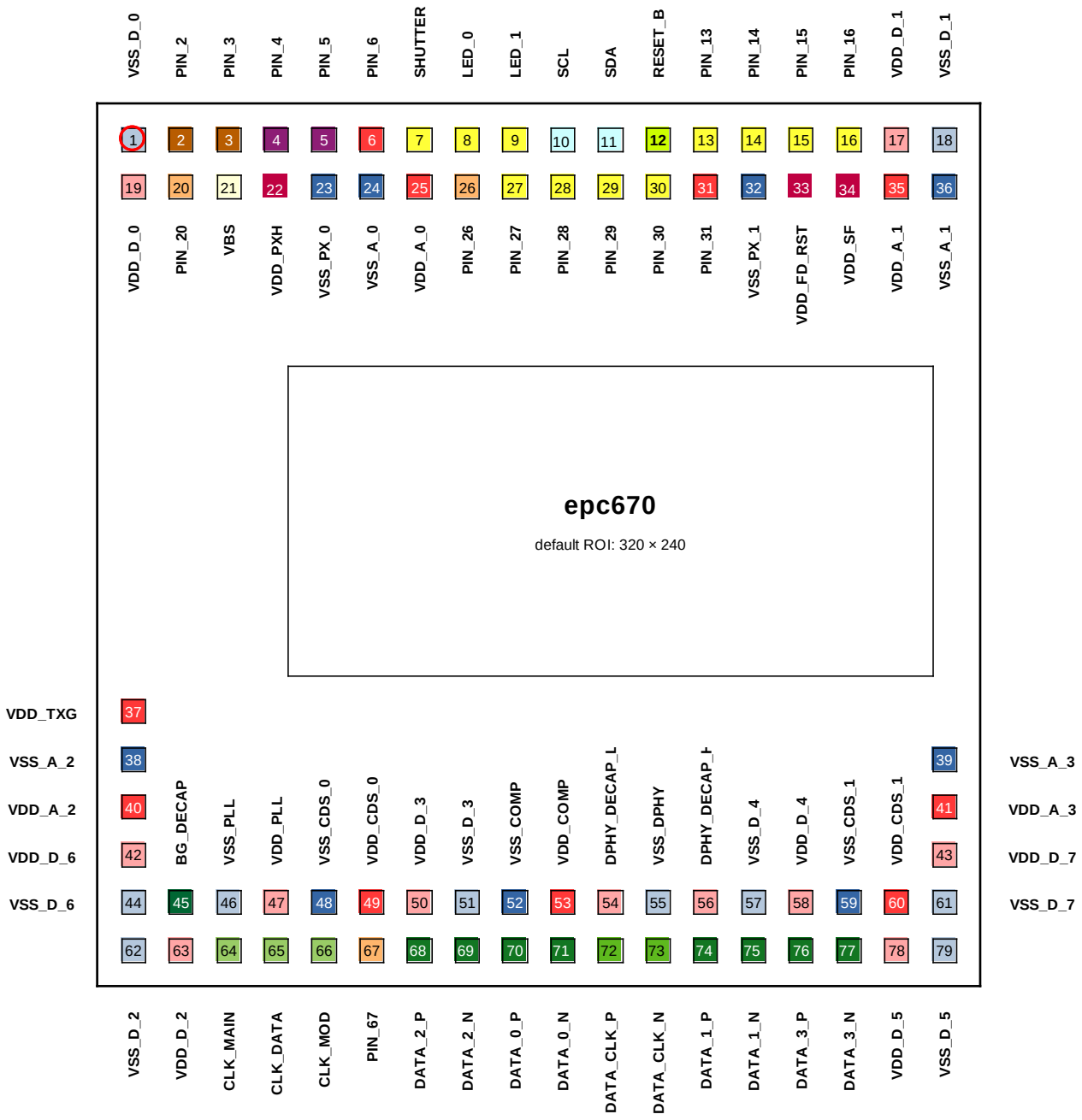

Figure 8: CSP pin mapping (top-view, solder balls are at the bottom, pixel field is at the top)

3.2. Pin list

Table 11: Pin list

Pin No.	Pad Name	Pad Type	Speed	State		Supply / Supply Domain	Description
				Reset	Active		
1	VSS_D_0	GND	Static			0.0 V	Digital core ground
2	PIN_2	AI	Static			10.0 V	Reserved, to be left open
3	PIN_3	AI	Static			10.0 V	Reserved, to be left open
4	PIN_4	AO				10.0 V	Reserved, to be left open
5	PIN_5	AO				10.0 V	Reserved, to be left open
6	PIN_6	PWR	Static			5.0 V	Reserved, to be left open
7	SHUTTER	SDI	10 MHz	IPD	IPD	1.8 V	Trigger signal for image acquisition
8	LED_0	SDIO	62.5 MHz	PD	O	1.8 V	LED push-pull pre-driver 0 (low current)
9	LED_1	SDIO	62.5 MHz	PD	O	1.8 V	LED push-pull pre-driver 1 (low current)
10	SCL	SDIO	1 MHz	I	IO	1.8 / 3.3 V	I2C clock (open drain)
11	SDA	SDIO	1 MHz	I	IO	1.8 / 3.3 V	I2C data (open drain)
12	RESET_B	SDI	Static	IPD	IPD	1.8 V	Asynchronous global reset signal (active-low, weak internal pull-down)
13	PIN_13	SDO	20 Mbps	Z	O	1.8 V	Reserved, to be left open
14	PIN_14	SDI	20 Mbps	IPD	IPD	1.8 V	Reserved, to be left open
15	PIN_15	SDI	20 MHz	IPD	IPD	1.8 V	Reserved, to be left open
16	PIN_16	SDI	20 Mbps	IPD	IPD	1.8 V	Reserved, to be left open
17	VDD_D_1	PWR	Static			1.8 V	Digital core power
18	VSS_D_1	GND	Static			0.0 V	Digital core ground
19	VDD_D_0	PWR	Static			1.8 V	Digital core power
20	PIN_20	AIO	Static			5.0 V	Reserved, to be left open
21	VBS	PWR	Static			-15.0 V	Pixel field backside bias voltage
22	VDD_PXH	PWR	Static			10.0 V	High voltage pixel driver power
23	VSS_PX_0	GND	Static			0.0 V	Pixel field ground (substrate)
24	VSS_A_0	GND	Static			0.0 V	Analog core ground
25	VDD_A_0	PWR	Static			5.0 V	Analog core power
26	PIN_26	AIO		Z		1.8 / 5.0 V	Reserved, to be left open
27	PIN_27	SDI	Static	IPD	IPD	1.8 V	Reserved, to be left open
28	PIN_28	SDI	Static	IPD	IPD	1.8 V	Reserved, to be left open
29	PIN_29	SDI	Static	IPD	IPD	1.8 V	Reserved, to be left open
30	PIN_30	SDI	Static	IPD	IPD	1.8 V	Reserved, to be left open
31	PIN_31	PWR	Static			5.0 V	Reserved, to be connected to 5.0 V
32	VSS_PX_1	GND	Static			0.0 V	Pixel field ground
33	VDD_FD_RST	PWR	Static			9.0 V	Floating diffusion reset power (sensitive to noise)
34	VDD_SF	PWR	Static			10.0 V	Source follower power (sensitive to noise)
35	VDD_A_1	PWR	Static			5.0 V	Analog core power
36	VSS_A_1	GND	Static			0.0 V	Analog core ground
37	VDD_TXG	PWR	Static			5.0 V	Transfer gate driver power
38	VSS_A_2	GND	Static			0.0 V	Analog core ground
39	VSS_A_3	GND	Static			0.0 V	Analog core ground
40	VDD_A_2	PWR	Static			5.0 V	Analog core power
41	VDD_A_3	PWR	Static			5.0 V	Analog core power
42	VDD_D_6	PWR	Static			1.8 V	Digital core power
43	VDD_D_7	PWR	Static			1.8 V	Digital core power
44	VSS_D_6	GND	Static			0.0 V	Digital core ground
45	BG_DECAP	AO	Static			5.0 V	Band gap reference voltage (1.15 V) for external decoupling capacitance (10 nF)
46	VSS_PLL	GND	Static			0.0 V	PLL ground
47	VDD_PLL	PWR	Static			1.8 V	PLL power
48	VSS_CDS_0	GND	Static			0.0 V	Analog CDS ground (sensitive to noise)
49	VDD_CDS_0	PWR	Static			5.0 V	Analog CDS power (sensitive to noise)
50	VDD_D_3	PWR	Static			1.8 V	Digital core power
51	VSS_D_3	GND	Static			0.0 V	Digital core ground
52	VSS_COMP	GND	Static			0.0 V	Analog ADC comparator ground (sensitive to noise)
53	VDD_COMP	PWR	Static			5.0 V	Analog ADC comparator power (sensitive to noise)
54	DPHY_DECAP_LP	PWR	Static			1.2 V	MIPI D-PHY low power mode power for external decoupling (10 nF + 1 µF)
55	VSS_DPHY	GND	Static			0.0 V	MIPI D-PHY ground
56	DPHY_DECAP_HS	PWR	Static			0.4 V	MIPI D-PHY high speed mode power for external decoupling (10 nF + 1 µF)
57	VSS_D_4	GND	Static			0.0 V	Digital core ground
58	VDD_D_4	PWR	Static			1.8 V	Digital core power
59	VSS_CDS_1	GND	Static			0.0 V	Analog CDS ground (sensitive to noise)
60	VDD_CDS_1	PWR	Static			5.0 V	Analog CDS power (sensitive to noise)
61	VSS_D_7	GND	Static			0.0 V	Digital core ground
62	VSS_D_2	GND	Static			0.0 V	Digital core ground
63	VDD_D_2	PWR	Static			1.8 V	Digital core power
64	CLK_MAIN	SDI	125 MHz	I	I	1.8 V	Clock input for main PLL
65	CLK_DATA	SDI	125 MHz	I	I	1.8 V	Clock input for PLL dedicated to data transmission (MIPI)
66	CLK_MOD	SDI	125 MHz	I	I	1.8 V	Modulation clock input
67	PIN_67	AIO		Z		1.8 / 5.0 V	Reserved, to be left open
68	DATA_2_P	DDOP	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 2 (positive)
69	DATA_2_N	DDON	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 2 (negative)
70	DATA_0_P	DDOP	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 0 (positive)
71	DATA_0_N	DDON	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 0 (negative)
72	DATA_CLK_P	DDOP	250 MHz	O	O	1.2 / 0.4 V	MIPI D-PHY differential clock lane (positive)
73	DATA_CLK_N	DDON	250 MHz	O	O	1.2 / 0.4 V	MIPI D-PHY differential clock lane (negative)
74	DATA_1_P	DDOP	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 1 (positive)
75	DATA_1_N	DDON	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 1 (negative)
76	DATA_3_P	DDOP	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 3 (positive)
77	DATA_3_N	DDON	500 Mbps	O	O	1.2 / 0.4 V	MIPI D-PHY differential data lane 3 (negative)
78	VDD_D_5	PWR	Static			1.8 V	Digital core power
79	VSS_D_5	GND	Static			0.0 V	Digital core ground

Notes:

[1] LED_0/1 outputs are push-pull drivers which deliver symmetric rise/fall times for an external illumination driver circuit. LED_0/1 may show slightly different insertion delays resulting in different distance results.

[2] RESET_B pin has a 600 kΩ (typical) internal pull-down resistor. Therefore, this pin can be safely connected to a standard GPIO of a CPU which is initially high-Z or open-drain during power up sequence. Once the SW takes control, it can program this GPIO as output and drive logic 1 to release the reset. The internal pull-down can be overridden by an external 10 kΩ pull-up and a series capacitor to build a simple delayed power-on reset for evaluation/qualification purposes.

[3] I^2C pins SCL and SDA are according to I^2C standard. They are I^2C slave pins which need external pull-up resistors on the PCB.

[4] If HW shutter is not used, connect the SHUTTER pin to GND.

'Pad Type' in Table 11 means the following:

- **AI**: **A**nalog **I**nput
- **AO**: **A**nalog **O**utput
- **AIO**: **A**nalog **I**nput/**O**utput (bidirectional)
- **SDI**: **S**ingle-ended **D**igital **I**nput
- **SDO**: **S**ingle-ended **D**igital **O**utput
- **SDIO**: **S**ingle-ended **D**igital **I**nput/**O**utput (bidirectional)
- **DDOP**: **D**ifferential **D**igital **O**utput **P**ositive
- **DDON**: **D**ifferential **D**igital **O**utput **N**egative
- **PWR**: Power
- **GND**: Ground

'State' in Table 11 defines the function of the IO pins during and after (active) reset:

- **I**: **I**nput
- **O**: **O**utput
- **IO**: **I**nput/**O**utput (bidirectional)
- **PU**: internal **P**ull-**U**p
- **PD**: internal **P**ull-**D**own
- **IPD**: **I**nput with internal **P**ull-**D**own
- **Z**: high impedance

Refer to Chapter 14 for hardware implementation details.

3.3. Power domain separation and ESD protection

The epc670 chip has various power domains and ground references which are interconnected and protected with ESD diodes as depicted in Figure 9. The diodes have a breakthrough voltage of 0.3 V. Designers have to make sure that none of these diodes become conductive either at power-up, power-down or during normal operation. VBS on pin 21 is directly connected to the isolation ring around the pixel field without ESD diodes.

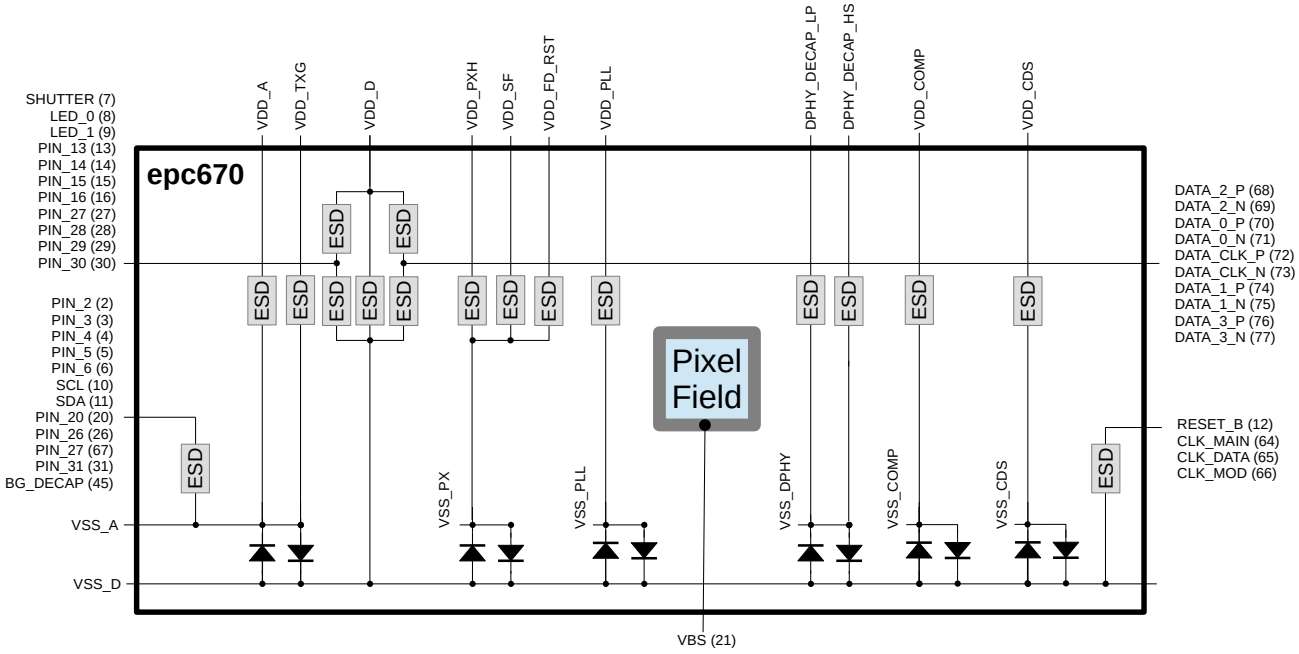

Figure 9: I/O pins and ESD protection diagram

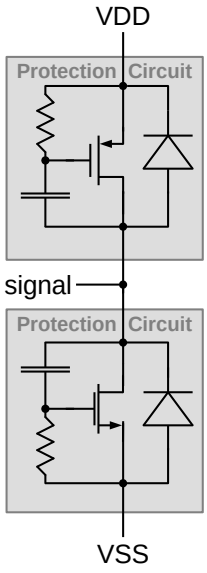

Figure 10: ESD protection circuits for power and signal

4. Packaging and layout information

4.1. Mechanical dimensions

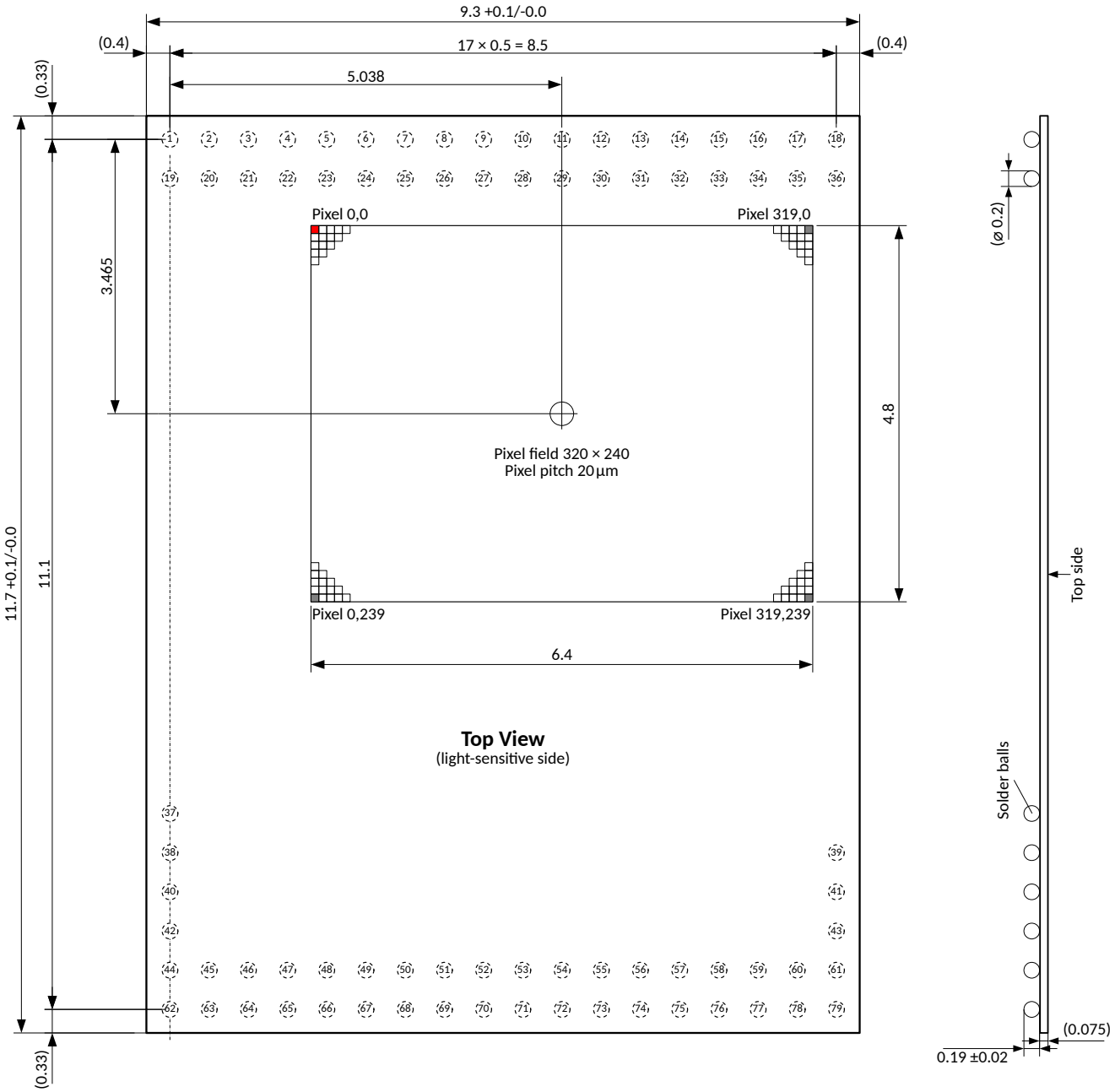

Figure 11: Mechanical dimensions (top view)

Notes:

- ⌖
- All measures are in millimeters.
- Not specified tolerances: ±0.001 mm
- Dimensions in brackets are informal only.
- Top side is illumination side.
- Pin 37 is asymmetrical by purpose in order to allow location identification with a vision system during the SMT assembly process. It is highly recommended to check the orientation of pins 1 and 37.

4.2. Location of the photosensitive area

The photosensitive area is not marked neither on the front nor on the backside of the IC. As a visible reference, a metal ring of the IC can be used. From the solder ball side, it is visible. Also from the front side (photosensitive area), it can be seen with a camera which is sensitive in the near infrared wavelength domain (950 nm to 1'150 nm).

4.3. PCB design and SMD manufacturing process considerations

As the epc670 chip comes in a 79 pin chip scale package with only 75 µm thickness, the PCB layout should be made with special care. In addition, careful handling during the assembly process shall be assured in order to avoid mechanical damage. Because the silicon chip is small and lightweight compared to the solder balls, it is highly recommended that all tracks to the chip should come straight from the side. A symmetrical design is highly recommended to achieve high production yield. The pads and the tracks should also have exactly the same width at least for 1 mm from the pads. The tracks shall be covered by a solder resist mask in order to avoid drain of the solder tin alloy along them.

A ground plane shall be placed on the top PCB layer underneath the chip. This ground plane is the common GND point and acts as a shield to suppress high frequency emission of fast interface signal lines. It is important that this plane is completely flat. Thus, the plane must not be scattered nor divided into sections. It should be rather full-faced and evenly plane for vias placed underneath this plane. Otherwise, chip bending might occur. In addition, the ground plane helps to dissipate the heat generated by the chip operation. A good heat dissipation is achieved if there is a temperature increase of the chip under normal operation of max. 20 K relative to the ambient temperature. Temperature data is available in the metadata of every image frame.

Underfill of the imager component reduces stress to the solder pads caused by e.g. temperature cycling or mechanical bending. Reducing the thermal and mechanical fatigue will increase the longterm reliability. Underfill material and underfill selection is application-specific. It shall follow JEDEC-STD JEP150: Stress-Test-Driven Qualification of and Failure Mechanisms Associated with Assembled Solid State Surface-Mount Components.

IMPORTANT:

Refer to the application note **AN08 Process-Rules CSP Assembly** which can be downloaded from the ESPROS website at www.espros.com, section downloads. Obeying these recommendations is very important to achieve a high manufacturing yield and high reliability.

4.4. Packaging information

The devices will be shipped in standard JEDEC trays for automatic placement systems. General tray specification data are available in a separate requirement specification. Further tray specifications can be found in the JEDEC Association standard JEP95.

The chips are not placed according to industry standard but as shown in Figure 12 (pin 1 not at the tray chamfer corner). ESPROS does not guarantee that there are no empty cavities. Thus, the pick-and-place machine should check the presence of a chip during picking. In addition, it should verify the correct location of pin 1.

The trays are designed for vacuum pick-up and for a maximum temperature of 150 °C.

Trays are packed and shipped in multiples of single trays with an empty cover tray on top. Trays are not a hermetic packaging. Therefore, the tray stack is sealed in a moisture barrier bag for storage and transportation.

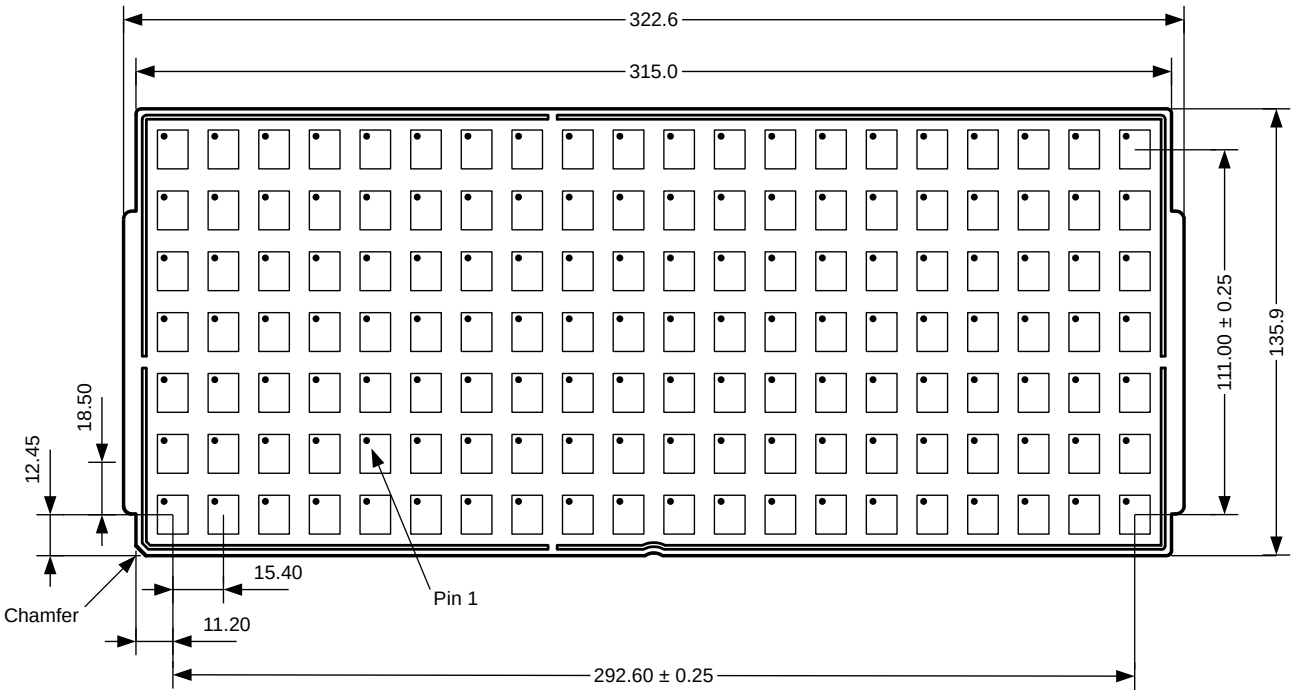

Figure 12: JEDEC tray for 7 × 20 pieces, maximum quantity of 140 pieces per tray, use vacuum pick-up (all measures in mm)

5. Supplies, resetting and start-up

5.1. Power-up sequence

At the beginning of the power-up sequence, VDD and VDD_PLL supplies (see Figure 13) must be applied at the same time to the epc670.

VDD_A, VDD_CDS, VDD_COMP and VDD_TXG supplies must be applied as a second group, after VDD_D and VDD_PLL supplies become stable.

VDD_PXH, VDD_SF and VDD_FD_RST supplies are recommended to be applied as a third group, after VDD_A , VDD_CDS, VDD_COMP and VDD_TXG supplies become stable.

The negative backside supply VBS is recommended to be applied after all positive supplies reached their rated levels or together with VDD_PXH, VDD_SF and VDD_FD_RST.

RESET_B must be kept low while all positive voltages are ramping-up in order to guarantee proper reset of all internal circuits. As soon as the rated positive levels are reached, RESET_B can be set to high. External clock applied on CLK_MAIN and CLK_DATA pins must be present before RESET_B is released. Releasing the reset then triggers the internal start-up sequence.

During the PLL locking time, the content of EEPROM page 0 is copied to the register bank. Several configuration registers are modified to have the default factory settings, i.e. the reset values are overwritten.

Furthermore, the content of EEPROM pages 1 to 4 is automatically loaded as fine trimming for the ADC gain of each individual ADC during t_{ADC_cal}.

IMPORTANT:

- The voltages for VDD_PXH, VDD_SF and VDD_FD_RST must not differ more than 5 V at any time.
- Image acquisition shall not start before all supply voltages are at their stable level.
- It is possible to shut down entire supplies for a very low standby current. In that case, RESET_B must be driven low first, then supplies can be turned off in the reverse order.
- VDD_A, VDD_CDS, VDD_COMP, VDD_TXG, VDD_PXH, VDD_SF and VDD_FD_RST supplies must never be kept on while turning off VDD_D and VDD_PLL. Damage to the chip can be the result.

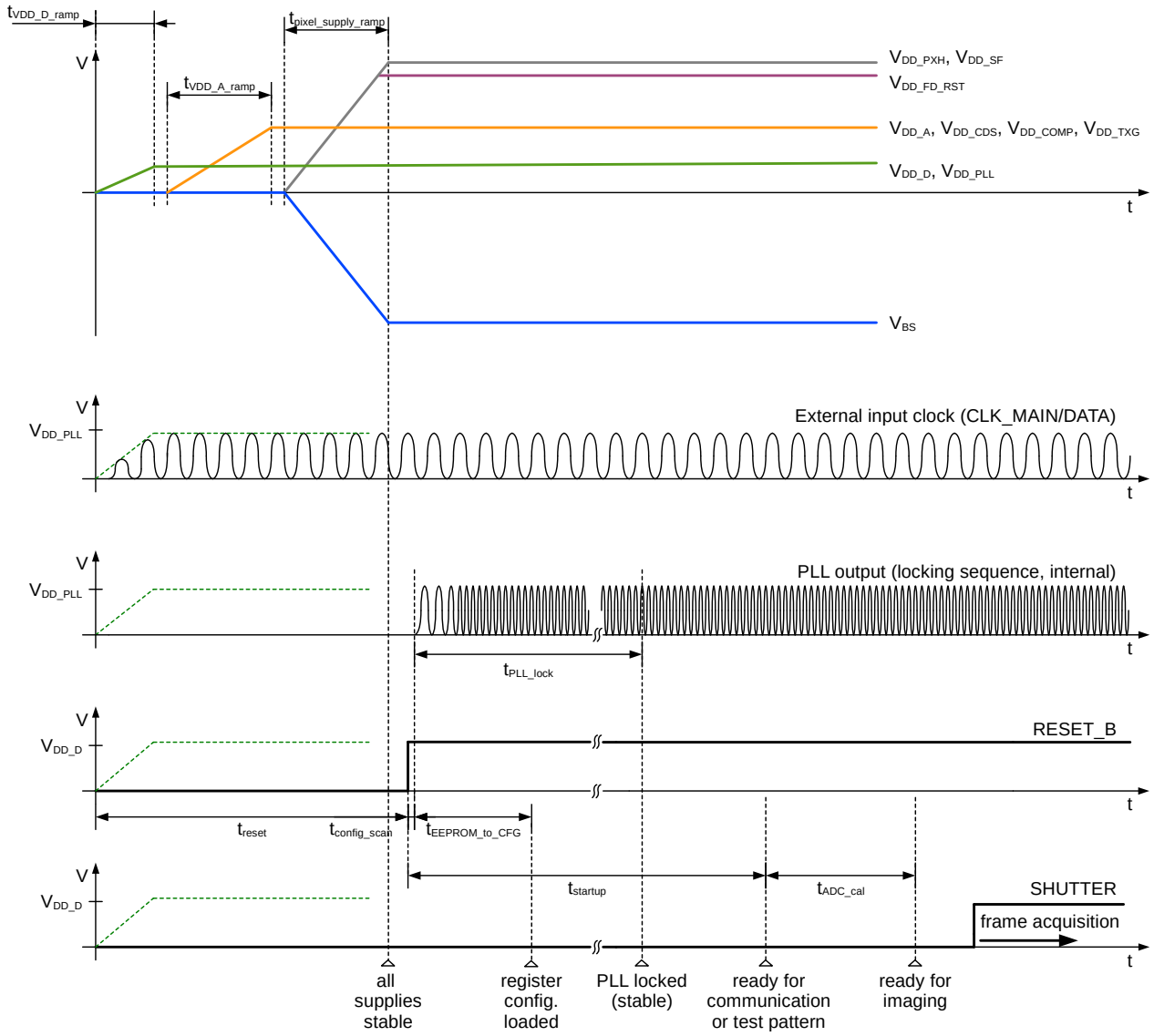

Figure 13: Power-up and reset sequence

5.2. Start-up and initialization sequence

1. Apply reset: RESET_B = 0.0 V
2. Apply all supplies (as described in Chapter 5.1).
3. Release reset: RESET_B = 1.8 V
4. Check that the readout sequencer is running (see Chapter 10.5.1).
5. Write readout sequencer program to volatile memory (see Chapter 10.5.2).

5.3. Power-down sequence

For powering down the epc670, the supplies shall be disabled in the reversed order compared to the power-up sequence (see Chapter 5.1).

6. Measurement control

6.1. Integration time setting

The integration time is the active frame acquisition period. Especially for moving objects or cameras, this time should be as short as possible to reduce or eliminate motion blur effects. The integration time together with the illumination intensity also defines the effective achievable operating distance.

The integration time can be set per DCS burst (measurement sequence triggered by a shutter event) by setting the 24-bit register field **integ_len** in registers INTEG_LEN_hi/mi/lo (addresses 0xA0 to 0xA2):

$$t_{integ} = \textbf{integ\_len} \times T_{clk\_mod}$$

For dual integration time mode (see Chapter 7.5.5), there is another register field **integ_len_dual** in registers INTEG_LEN_DUAL_hi/mi/lo (addresses 0xA3 to 0xA5) which controls the integration time for the odd rows:

$$t_{integ\_dual} = \textbf{integ\_len\_dual} \times T_{clk\_mod}$$

Alternatively, the integration time can be pre-configured per DCS in the frame configuration table.

6.2. Single measurement control

Individual measurements (four DCSs by default) can be triggered either by driving the SHUTTER pin high for at least 100 ns (HW shutter) or by setting a dedicated register flag (SW shutter). The selected measurement mode (4 DCSs, 2 DCSs, grayscale, etc.) defines how many frames the chip performs by the stimulation of one SHUTTER pulse for a measurement cycle.

While the SW controlled shutter is automatically cleared after propagation, the HW shutter must be set back manually latest before the end of the data transmission of the last DCS frame. During such a measurement cycle, the next frame acquisition (DCS) starts immediately after the last data readout on the MIPI until all frames are performed.

6.3. Continuous measurement control (video mode)

Alternatively to trigger single measurements, it is possible to enable the continuous measurement mode by keeping the SHUTTER pin high or setting a dedicated register flag. In continuous measurement mode, distance measurements are performed at maximum frame rate, i.e. as soon as a distance measurement is completed another one is started.

6.4. Acquisition timing

Figure 14 depicts the timing around the integration phase. There are two synchronization stages which slightly delay (less than 100 ns) the start of the modulation sequence with respect to the shutter pulse. t_{mod_delay} is kept minimum but can be adapted in the readout sequencer program in case powering up the readout circuitry needs more time (not relevant for long integration times). The purpose of the preheat phase is to preheat the LED and to prepare the flushing of the pixel field. The length of the preheat phase $t_{preheat}$ is configurable. Also the length of the flush phase t_{flush} is configurable. t_{flush} can be reduced to less than 50 ns. The integration phase is the period during which the pixels are collecting photo-induced charge. Its length is configurable, too. The integration phase is followed by the readout phase.

The time base for preheat and integration time settings is defined by the modulation clock frequency f_{CLK_MOD} or rather its corresponding period T_{CLK_MOD}. A 24-bit register allows integration times t_{integ} with continuous modulation up to more than 400 ms for f_{CLK_MOD} = 20 MHz.

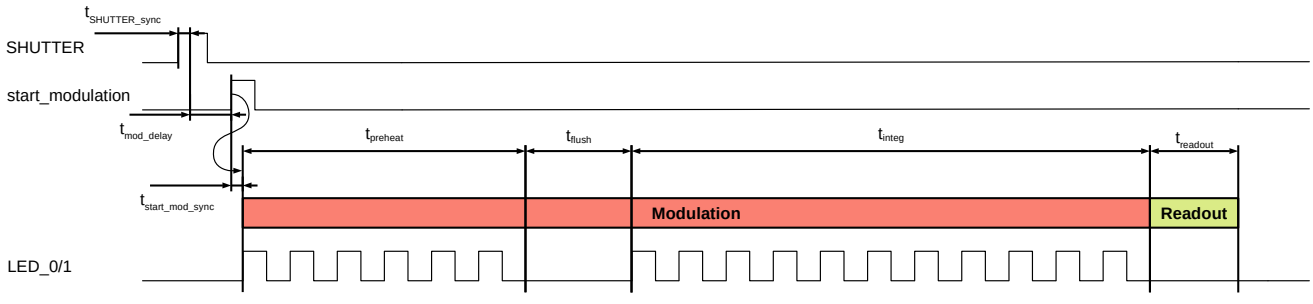

Figure 14: System level timing around integration phase

Note: The timing in Figure 14 is not in scale. The readout phase is often longer than the modulation phase.

7. Imaging

7.1. Pixel architecture

The pixels are organized in groups 2×2 pixels, called herein “pixel group”. The pixel group performs two basic operations: Measurement (integration) and readout (conversion of charge to digital value). Pixels are named as UE (Upper-row, Even-column), UO (Upper-row, Odd-column), LE (Lower-row, Even-column) and LO (Lower-row, Odd-column) depending on their location within the pixel group (see Figure 15). Pixels with the same name are controlled simultaneously in the whole pixel-field.

The pixel group architecture allows the epc670 to operate the pixel field in different modes and in combinations thereof as described in the following chapters.

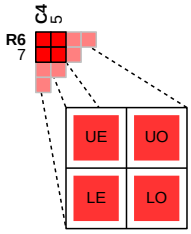

Figure 15: Pixel group consisting of four pixels

Each pixel of a pixel group has its own pair of storage gates SGa and SGb. During an integration phase, they accumulate charges (e-) excited by photons of modulated light reflected from objects in the scenery. The charge flow is controlled by the modulation gates MGa and MGb (demodulation). When an integration phase is finished, a readout phase starts. The charges stored in the storage gates SGa and SGb are read out in parallel over two separate source followers. As a next step, correlated double sampling (CDS) is done in a differential way. The CDS output is converted into a 12-bit digital value. Additionally, a 1 bit saturation flag to indicate overfilled storage gates is generated. Common mode (red parts, e.g. due to ambient light) is automatically suppressed during the parallel readout of SGa and SGb. This is illustrated in a simplified way in Figure 16.

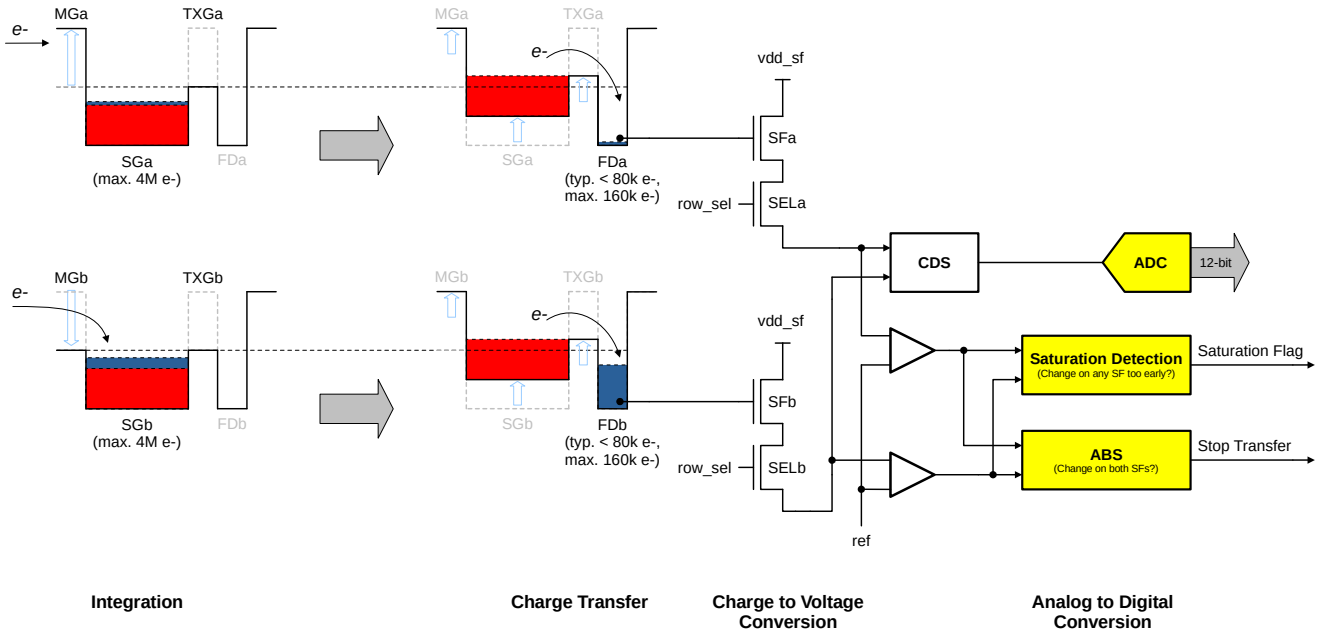

Figure 16: Simplified function overview

7.2. Pixel saturation detection

The pixels collect continuously modulated and non-modulated ambient light during the integration period. Depending on these light intensities, the pixels may collect more charge than they can accommodate (over-exposure) in their storage gates (refer to Figure 16). In such a case, the 12 bit sample data is not valid and cannot be used for distance calculation.

Each pixel generates a “saturation detection” flag along with the sample data, so that the data can be discarded by the application. In case of pixel saturation, the 12 bit data is set to 0x000 (unsigned) or 0x800 (signed), i.e. to the same value which is also indicating ADC underflow.

7.3. Signal chain

The signal chain is depicted in Figure 17. The section with yellow background is the pixel whereas the remaining parts are shared by all pixels connected to the same column output bus.

Photons intrude the chip from the backside. If they are absorbed in the fully depleted bulk, they free an electron and a hole. The hole drifts to the backside of the chip. The electron drifts to the most positive potential at the front side of the chip which is inside the pixel. This process is illustrated as a diode in Figure 17. From there, depending on the gate potential at MGa or MGb the electron drifts further either through the modulation gate MGa or MGb to the storage gate SGa or SGb, respectively. The photoinduced charge is then accumulated at the storage gates which is called integration. The duration of the integration is called integration time t_{integ}. Once the integration is completed, the charge at the storage gates selected for readout is simultaneously drained through the transfer gate TXGa and TXGb to the floating diffusion FDa and FDb. The floating diffusions are like capacitors and convert charge to voltage. The gate capacitance of the source follower structure contributes as well to this capacitance.

The draining process is controlled by continuously monitoring the voltage at the source follower output. It is immediately stopped when the storage gate with the smaller amount of charge starts generating a signal at the corresponding source follower output. As a result, on one floating diffusion is the difference between SGa and SGb plus a small amount of charge spilled over until the draining process is stopped. On the other floating diffusion is just the spilled-over charge. This spilled-over charge is denoted as Stop Offset in Figure 17.

These two amounts of charge are converted to voltage and the resulting voltage difference at the outputs of source followers SFa and SFb is amplified by a selectable gain at the differential CDS stage. The Stop Offset is canceled at the CDS stage due to its differential architecture. However, too much spilled-over charge arriving at the floating diffusions may result in distance errors when the limit of the FD capacity is reached. Therefore, it is important that the reaction time to stop the charge draining process is minimum.

Finally, the CDS output is converted to an unsigned 12-bit digital number at the ADC.

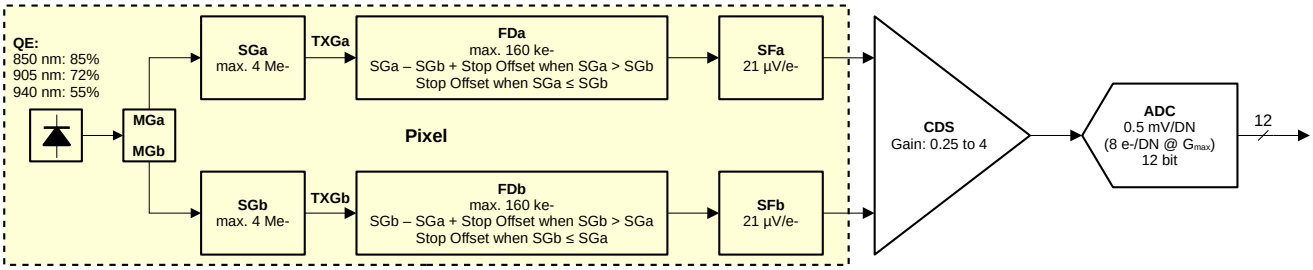

Figure 17: Analog signal chain

Notes: All figures are typical values.

7.4. Pixel coordinates

The epc670 pixel-field consists of a total of 328 × 256 pixels of which 320 × 240 are active by default. 8 rows top/bottom and 4 columns left/right on the periphery of the pixel-field contain dummy pixels. The upper-left corner (top view on chip) is the origin (4/8) of the epc670 pixel-field. X-axis starts at 4 and counts up to 323 to the right. Pixel y-axis starts at 4 and counts up to 247 to the bottom. All readout modes and control registers use this coordinate system to set or change modes of the chip.

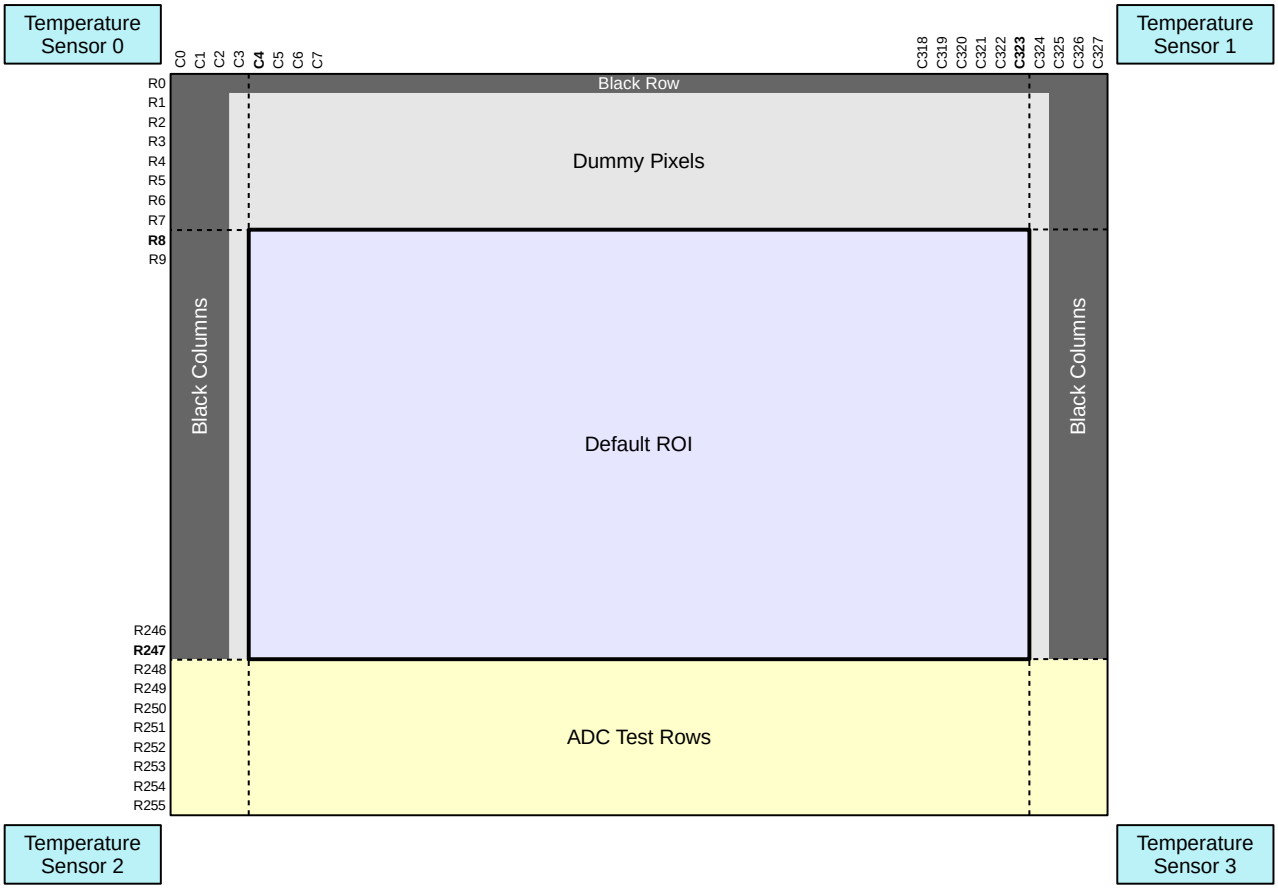

Figure 18: Pixel-field coordinates with row and column numbering scheme (top-view, solder balls on bottom side)

7.5. Pixel field operation modes

7.5.1. Full resolution mode (default)

This is the default operation mode for 3D TOF operation. All UE, UO, LE, LO storage gates work simultaneously during measurement operation. The storage gate control signals mga, mgb are applied to all pixels simultaneously (see Figure 19). Up to 16 DCSs per shutter event can be acquired in this mode – optionally with individual settings from the frame configuration table for each DCS.

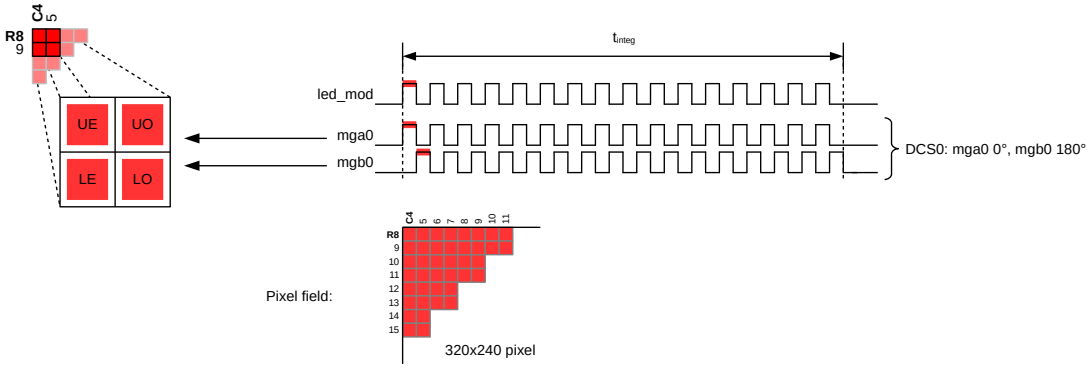

Figure 19: Full resolution mode

7.5.2. Pixel binning

The charges accumulated in the storage gates during integration can be combined by binning. Binning can be enabled horizontally, vertically or in both directions (horizontal+vertical; see Figure 20). Advantages of binning are higher sensitivity, reduced integration time and faster readout of frames.

IMPORTANT:

- Vertical binning cannot be used together with dual phase or dual integration time mode.
- Binning should not be used in combination with automatic ambient light suppression, i.e. ABS should be disabled in that case.

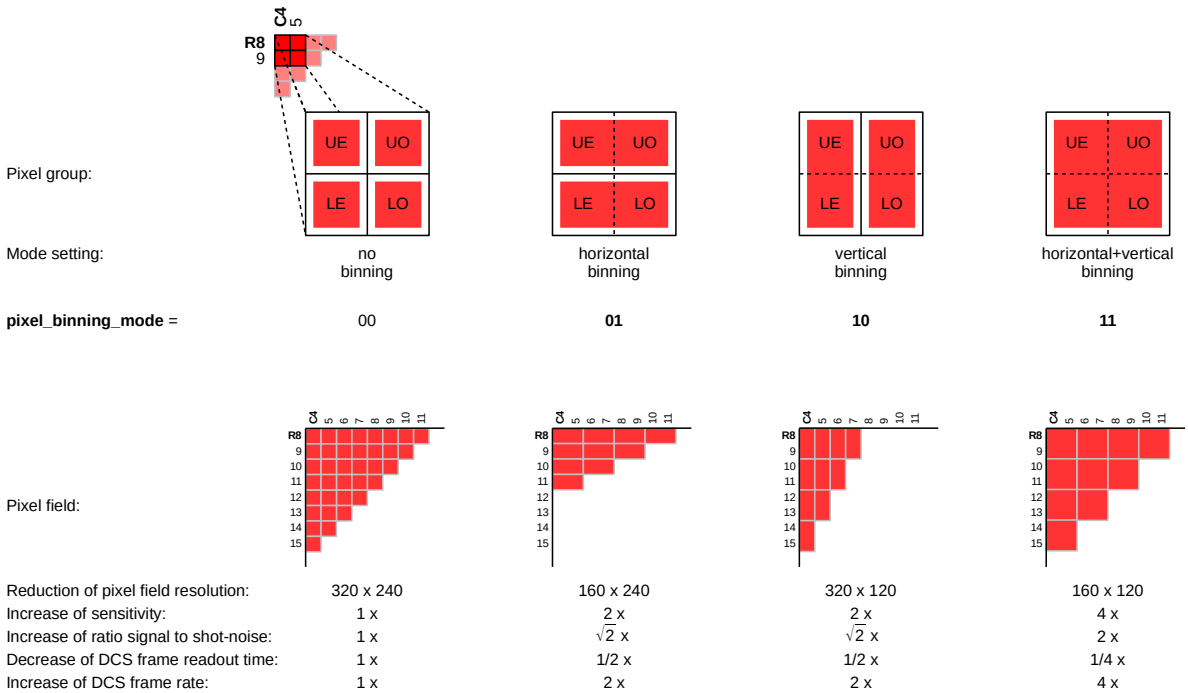

Figure 20: Pixel binning modes and readout

7.5.3. Resolution reduction

Resolution reduction, where only every second pixel is read out horizontally, vertically or both, is supported as an independent feature. It can be combined with binning (see Chapter 7.5.2), ROI (see Chapter 7.5.6), motion blur reduction and high dynamic range modes. Figure 21 depicts the four basic resolution reduction modes.

Resolution reduction shrinks the dataset to the necessary amount of data required for the application. The advantages are the reduced amount of data to be processed for the final measurement result (reduced frame buffers) and the faster processing (shorter readout and processing time).

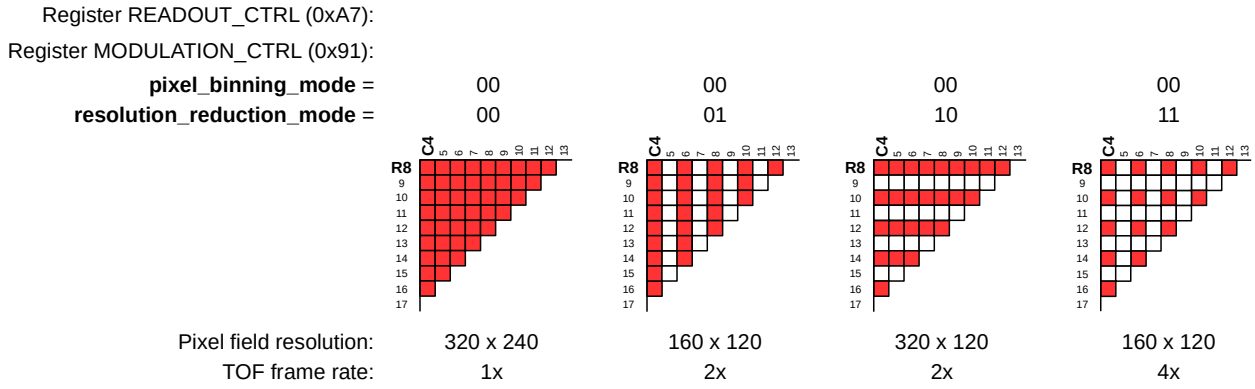

Figure 21: Resolution reduction

7.5.4. Dual phase mode (motion blur reduction)

In this mode, the even and the odd rows are controlled by phase shifted signals (e.g. by 90°, see Figure 22). This mode allows to acquire two DCSs at the same time, e.g. DCS0 and DCS1. In the 2 DCS mode, distance calculation can be accomplished within one acquisition. Thus, motion blur is eliminated. The even row pixels store DCS0 (or DCS2) while the odd row pixels store DCS1 (or DCS3). The vertical pixel pairs (e.g. UE/LE) must be treated for distance calculation as if they are one single pixel. This comes at the cost of a reduced resolution along the y-axis. The result provides a total of 320 × 240 pixel field readout with an effective 3D TOF resolution of 320 × 120 pixels.

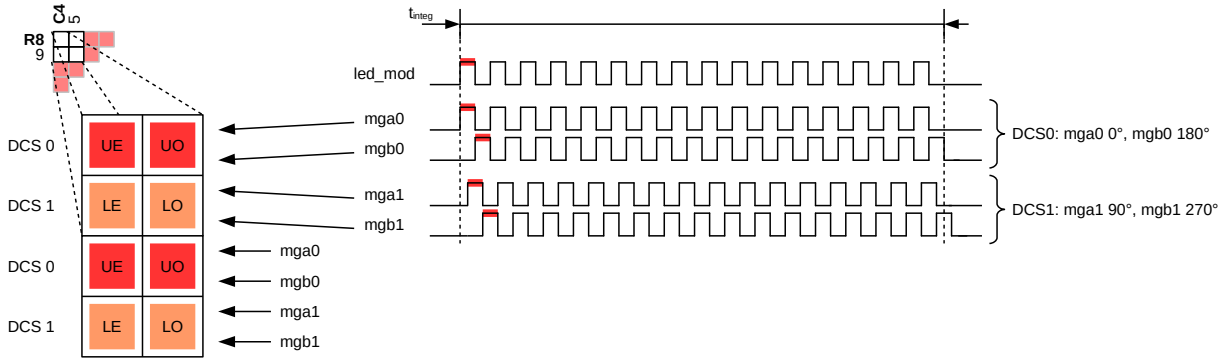

Figure 22: Dual phase mode

IMPORTANT: This mode requires that adjacent pixels look to the same point on the target and receive the same amount of light. Otherwise, calculated distance values are not reliable. A pixel with a large offset or a defective pixel will lead to completely wrong distance values with its paired pixel. Therefore, the group of pixels must be discarded.

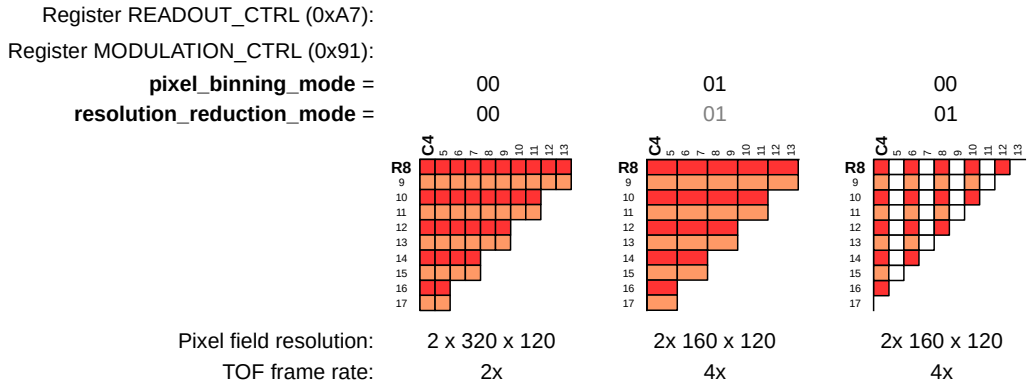

Figure 23: Regular dual phase mode with binning and row reduction

7.5.5. Dual integration time mode (high dynamic range, HDR mode)

In this mode, the even and the odd rows are controlled with different integration time lengths. This allows to acquire one image with two different integration times in order to increase the dynamic range. Both groups provide exactly the same DCS modulation signals (phases). One stops earlier than the other due to different integration times (see Figure 24). As a consequence, the two pixels collect different amount of light simultaneously. There is no restriction about which integration time is shorter or longer with respect to the other. The even row pixels integrate with integration time t_{integ} while the odd row pixels integrate with integration time t_{integ_dual}. The even and odd pixels (e.g. UE, UO) must be used independently for distance calculation. Finally, the vertical pixel pairs (e.g. UE/LE) must be treated as if they are one single pixel by using only the better of the two pixel signals. This comes at the cost of a reduced resolution along the y-axis. Instead of one frame with 320 × 240 pixels, a single readout provides two DCSs or black and white frames with an effective resolution of 320 × 120 pixels but with different integration times.

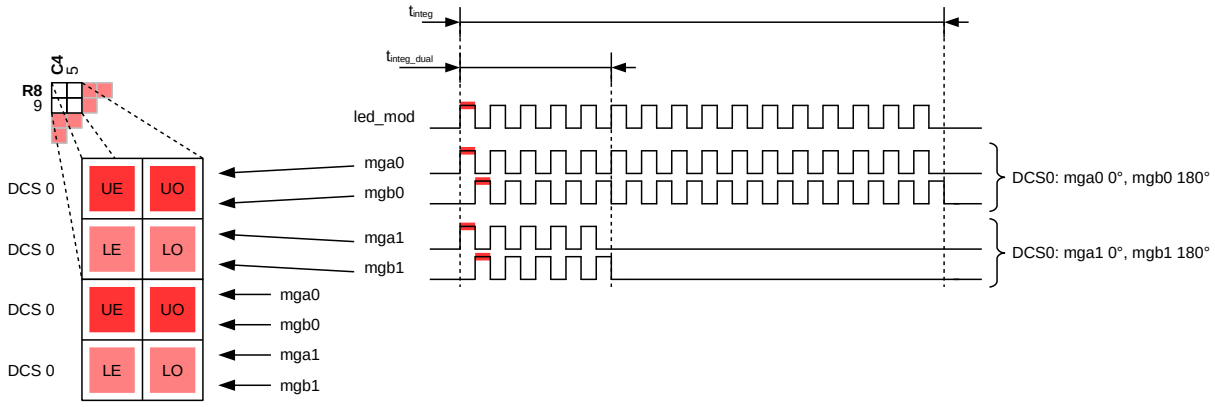

Figure 24: Dual integration time mode

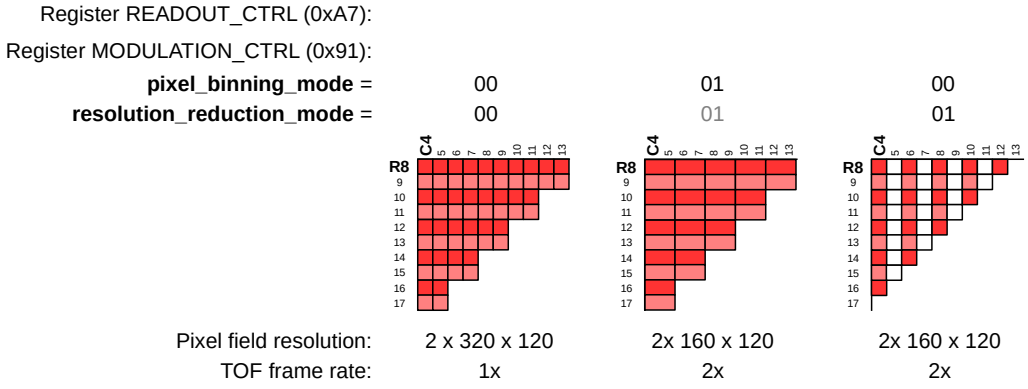

Figure 25: Regular dual integration time mode with binning and row reduction

7.5.6. Region of interest (ROI)

The ROI feature allows to read out and transfer the portion of the pixel-field data which is necessary for an application. The advantage of a smaller ROI is that less data needs to be read out and processed which allows higher frame rates. For integration times in the µs range, much shorter than the row conversion time (see Table 8), the frame rate scales with the set number of rows of the ROI.

The ROI feature is always active. The ROI is defined horizontally and vertically by start and end columns and rows (see Figure 26), respectively. The ROI starts with even row and column and ends with odd row and column. Additionally, vertical start and end indices must be multiples of eight (**roi_v_start** = n_{start}×8) and multiples of eight minus one (**roi_v_end** = n_{end}×8-1), respectively. Start indices always need to be smaller than end indices (n_{start} < n_{end}).

Pixels in the selected ROI are read out from top to bottom, i.e. from lowest to highest row index within ROI.

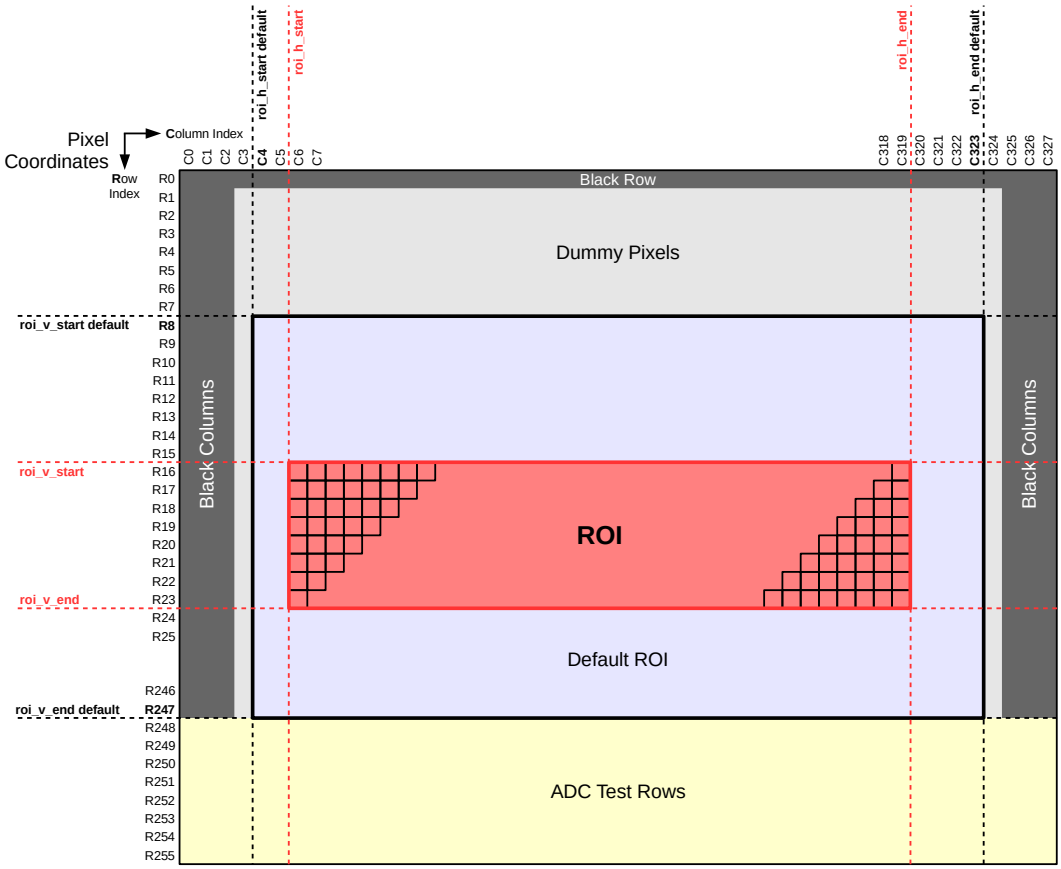

Figure 26: Region of interest (ROI)

The ROI registers can be changed on-the-fly via I²C all the time. The new values will be used starting with the next shutter event. The application must use the same ROI during the data readout.

IMPORTANT:
1. ROI can be set to a minimum rectangle of columns by rows of 4 by 8.
2. If horizontal reduction is enabled, ROI can be set to a minimum rectangle of columns by rows of 8 by 8.

7.6. Distance measurement (3D TOF)

The epc670's default modulation mode is based on the sinusoidal TOF modulation theory but uses effectively for the illumination a square-wave modulated signal with a fixed duty cycle of 50%. After reset, all internal register values are default to operate the chip at 80 MHz external modulation clock input. The modulation clock must be provided separately to pin CLK_MOD. Its frequency can be doubled with a built-in frequency doubler or optionally divided internally. For 200 MHz modulation clock at the CGU output, 50 MHz LED/LD modulation are obtained. By default, 4 successive DCS frames (0 ... 3) are acquired.

The distance measurement mode uses the on chip LED driver and the external LED/LD to provide modulated light on the target. Modulation control signals to the LED driver are provided by a programmable modulator. The modulator generates all signals to modulate the external LED/LD and simultaneously all demodulation signals to the pixel-field. TOF and grayscale mode with all the variants are generated here.

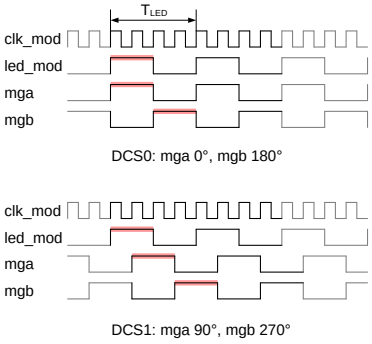

Figure 27: 4 DCS modulation/demodulation waveforms

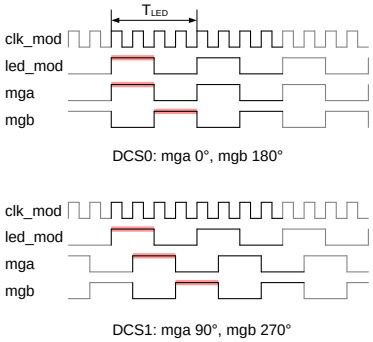

Figure 28: 2 DCS mod./demod. waveforms

Bytes 15, 14 and 13 of a frame configuration table (FCT) entry (see registers FCT_DATA_15, FCT_DATA_14 and FCT_DATA_13) control the modulation. The frame configuration table can be updated via I²C bus at any time. Updated settings will become effective starting with the next DCS burst.

With the application of the shutter pulse (HW SHUTTER or SW shutter via I²C), the chip performs the required number of successive DCS acquisitions. Each one of the 4 DCS frame types has a different phase relation between modulation (led_mod) and demodulation (mga, mgb) signals which makes phase-to-distance calculation possible. In case of DCS0, led_mod is phase-shifted by 0° and 180° with respect to mga and mgb, respectively. In case of DCS1, led_mod is phase-shifted by 90° and 270°. For DCS2, the phase shifts are 180° and 0° and for DCS3, the phase shifts are 270° and 90° (see Figure 27). Note that for DCS2 and DCS3, the demodulation signals mga and mgb are simply swapped with respect to DCS0 and DCS1, respectively.

By programming the number of DCS readouts to 2 (see **dcs_num_sel** of register SHUTTER_CTRL), a shutter initiates 2 successive DCS frame acquisitions (see Figure 28). This mode allows distance acquisition by using two DCSs only and thus a doubled frame rate. However, the cost is a lower distance measurement accuracy and a 40% higher distance noise.

7.7. Distance calculation algorithm

The use of the trigonometric atan2 definition for vectors (x, y) in the Cartesian coordinate system φ = atan2(x, y) = atan2(y/x) guarantees a continuous distance calculation algorithm in the range of phases between -π … +π. In our case, we use the range from 0°… 360° which corresponds to the distance from 0 m up to the unambiguity distance (refer to Figure 29 and Figure 30).

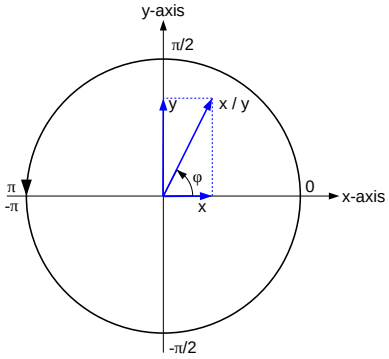

Figure 29: Continuous atan2 representation for the range -π … +π

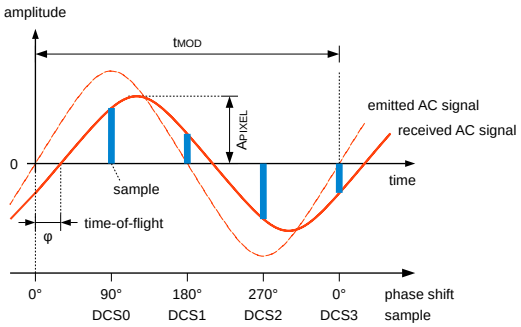

Figure 30: Sampling of the received waveform

Typically, the distance is calculated by using the 4 DCSs, also called π-delay matching, which cancels pixel offsets leading to distance errors:

$$[1] \quad D_{TOF}[m] = \frac{c}{2} \cdot \frac{1}{2\pi f_{LED}} \cdot \left[\pi + \operatorname{atan2}\left(\frac{DCS3 - DCS1}{DCS2 - DCS0} \right) \right] + D_{OFFSET}$$

The measured data are always over the 360° phase-shift valid. Due to the distance offset adjustment D_{OFFSET}, the correction of the distance roll-over effect at zero and unambiguity distance is necessary for having all the time correct distance values D:

- if $D_{TOF} > D_{unambiguity}$: $D = D_{TOF} - D_{Unambiguity}$
- if $D_{TOF} < 0$: $D = D_{TOF} + D_{Unambiguity}$
- else: $D = D_{TOF}$

If higher distance errors can be tolerated but a high frame rate is needed, the distance calculation also works with 2 DCSs only. Note that the data format of the DCS values has to be signed for the following formula to be correct:

$$[2] \quad D_{TOF}[m] = \frac{c}{2} \cdot \frac{1}{2\pi f_{LED}} \cdot \left[\pi + \operatorname{atan2}\left(\frac{-DCS1}{-DCS0} \right) \right] + D_{OFFSET}$$

The following terms are used in the formulas above:

D_{TOF}	Distance in meters [m]
c	Speed of light 299'792'458 [m/s]
f_{LED}	LED/LD modulation frequency (e.g. 12MHz) [1/s]
DCS0 - DCS3	Sampling amplitude [LSB]
φ	Phase shift caused by the time-of-flight [rad]
D_{OFFSET}	Offset compensation [m]
$D_{Unambiguity}$	Unambiguity distance [m]

7.7.1. Unambiguity range versus time base setting

Due to continuous modulation, roll-over can be observed if the distance to the object is longer than the length of one modulation cycle (one period, 2π). This roll-over distance is called unambiguity range can be calculated as follows:

$$[3] \quad D_{Unambiguity}[m] = \frac{c}{2} \cdot \frac{1}{f_{LED}}$$

The operating range is the maximum distance which corresponds to the maximum time-of-flight inside of one period of the used modulation: It is one period of f_{LED}. Objects inside this area are detected unambiguously.

The unambiguity range defines the repetition distance, where objects outside of the targeted operating range can still be detected as far they are of very high reflectivity (remission). Strongly reflected signals outside of this range may therefore interfere with the measurement.

The operating range, the unambiguity distance, the time base for the integration time and the resolution of the distance signal are defined by the modulation clock CLK_MOD. This corresponds for the epc670 to a maximum default operating range of 15 m @ f_{clk_mod} = 10 MHz. It may be necessary depending on the application to adapt these parameters to other values. It can be done by a change of the modulation clock. Table 17 lists as an example some values of the modulation clocks in function of the unambiguity distance, of the distance resolution and of the multipliers of the integration time base.

7.7.2. Confidence data

The DCS values contain not only the distance information but also the confidence level of the received optical signal. The higher the signal amplitude of the received signal, the better and more precise the distance measurement. Each distance measurement of every pixel has its own validity and quality.

The primary quality indicator for the measured distance data is the amplitude of the received modulated light A_{TOF}. The amplitude is in direct relationship to the distance noise (refer to Figure 57). The amplitude can be calculated as follows:

$$[4] \quad A_{TOF} = \frac{\sqrt{(DCS2 - DCS0)^2 + (DCS3 - DCS1)^2}}{2}$$

A simplified calculation method is simply to add the absolute values of all DCSs (signed format):

$$[5] \quad A_{TOF\ simplified} = |DCS0| + |DCS1| + |DCS2| + |DCS3|$$

This method is much faster and has the benefit that at phase angle positions (PHI), where the accuracy is lower due to values close to zero, more signal is required to obtain the same distance noise level as at phase angle positions 45°, 135°, 225°, and 315°. Figure 31 shows the resulting amplitude values.

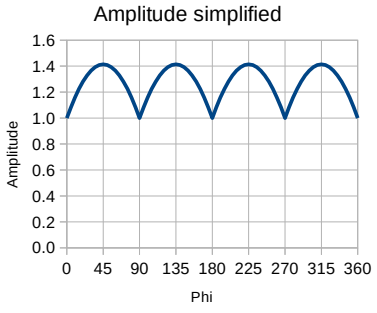

Figure 31: Simplified amplitude calculation values (normalized)

Table 12: ToF amplitude quality (CDS gain of 1.5)

Amplitude A_{TOF}	Classification	Action
< 40 LSB	Weak signal	Objects can be detected but distance measurement is highly inaccurate. Increase the integration time.
40 … 100 LSB	Distance measurement possible	High distance noise, increase the integration time.
100 … 2'000 LSB	Good signal strength	No action necessary
> 2'000 LSB	Overexposed	Decrease integration time.

Notes:

- The ranges for small ToF amplitudes listed in Table 12 depend on the selected CDS gain. The higher the CDS gain, the more ToF amplitude is required to achieve good signal strength.
- The amplitude value is the feedback parameter that is used to set the integration time for the next measurement. Generally, the higher the received signal, the better and more precise the distance measurement. However, it is good practice to control the integration time such that an amplitude value between 100 ... 2'000 LSB is achieved.

7.8. Grayscale imaging

The grayscale mode allows using the epc670 as a grayscale imager. This mode can be used either without LED/LD illumination for ambient light measurements or with LED/LD for active illumination of the scenery (e.g. at night). The readout of the pixel field is also performed differentially like in ToF mode. Just the integration happens only over MGa (permanently on during integration). When using signed data format, the 12-bit image data is in the range of ~0 LSB to 2'047 LSB. Note that the starting point can vary depending on ADC gain trimming and DSNU offset and can even be negative. Furthermore, the saturation flag status is invalid in this mode.

Due to fact that distance measurement results can be influenced by ambient light, grayscale measurements without illumination can be used as an important quality and correction parameter for distance measurements.

The irradiance E_{BW} at the surface of a pixel corresponding to received grayscale image data can be calculated from the DC sensitivity S_{BW}, the used integration time t_{integ_BW}, the reference integration time $t_{integ\_BW\_ref}$ and the signal level of the grayscale image as follows:

$$[6] \qquad E_{BW} = S_{BW} \cdot \frac{t_{integ\_BW\_ref}}{t_{integ\_BW}} \cdot \frac{G_{max}}{G} \cdot BW\_DATA \qquad \text{e.g.} \quad E_{BW} = 90 \frac{pW/mm^2}{LSB} \cdot \frac{100\,\mu s}{1000\,\mu s} \cdot \frac{4}{1.5} \cdot 1'000\,LSB = 24 \frac{nW}{mm^2}$$

8. Functional safety

8.1. Overview

The epc670 chip offers test features which allow to test the imager functionality during runtime. Blocks providing such functional test features are highlighted with red color in Figure 32.

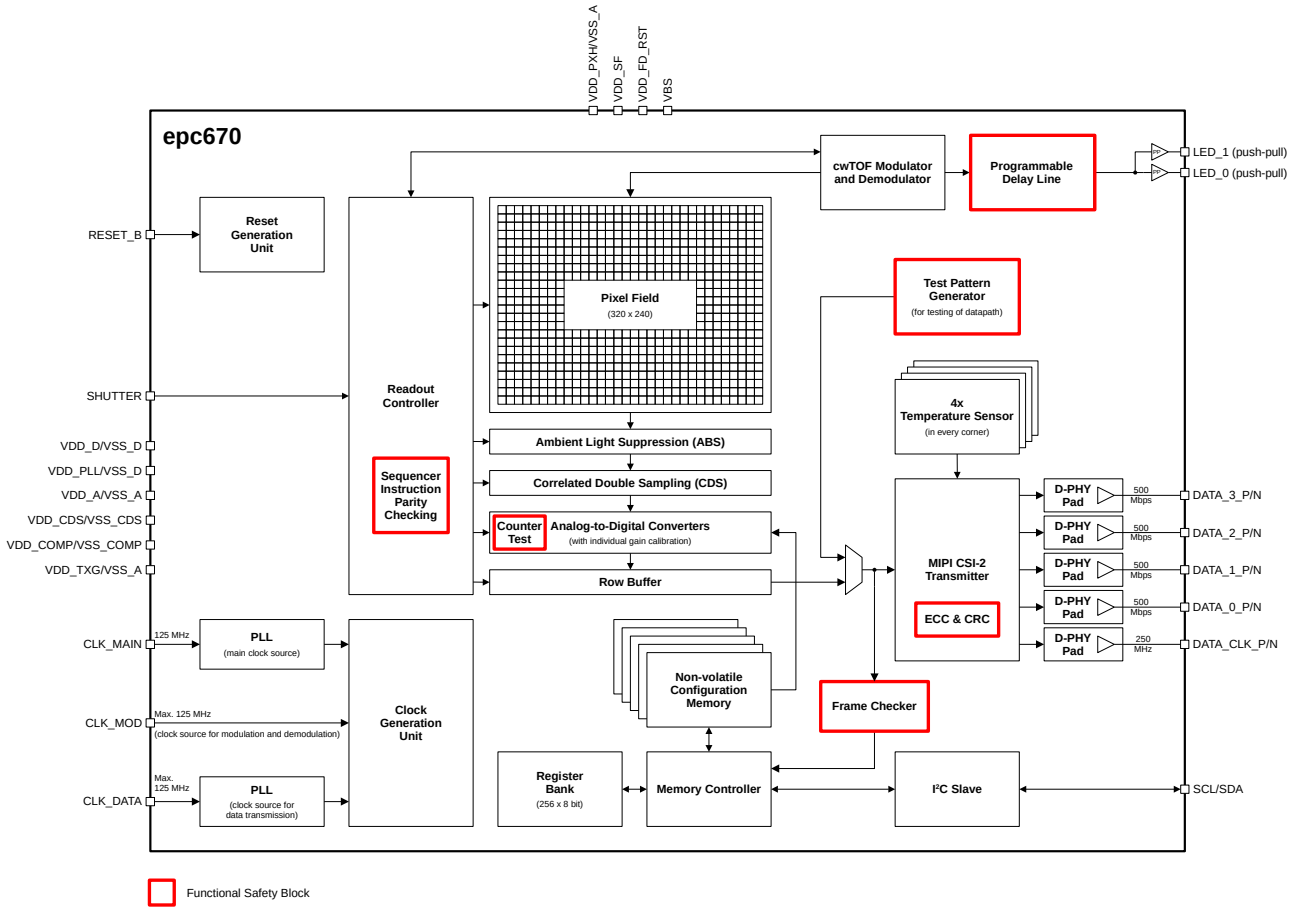

Figure 32: Block diagram with marked functional safety test (FST) blocks

8.2. Test images

Frames containing predefined digital test patterns can be triggered to test the digital data path starting at the input of the MIPI CSI-2 controller. This can be very helpful for debugging during system integration. Furthermore, such tests can also be done from time to time during operation to check whether the data transmission of the imager is operating normally, i.e. built-in self-tests (BIST) can be implemented on system level.

The supported test patterns can be selected using register field **fst_mode_sel**. They are described in Chapters 8.2.1, 8.2.2 and 8.2.3. Ramp and PN9 test patterns are computed based on the pixel index. The pixel indexing for a certain ROI (see Chapter 7.5.6) is explained in Figure 33. Additionally, a frame index is added for ramp and PN9 test patterns. The frame index is a 16-bit value starting at 0 and incrementing with each frame (DCS). When the 16-bit limit is reached, it simply wraps around and starts incrementing again. The frame index can also be reset manually using register bit **reset_frame_number**.

Test images are triggered by setting register bit **fst_trigger** which is automatically cleared after the transmission of a test pattern frame.

Note: There will neither be an integration (also no LED modulation) nor a pixel field readout phase for test images, i.e. only the transmission of digital data is triggered.

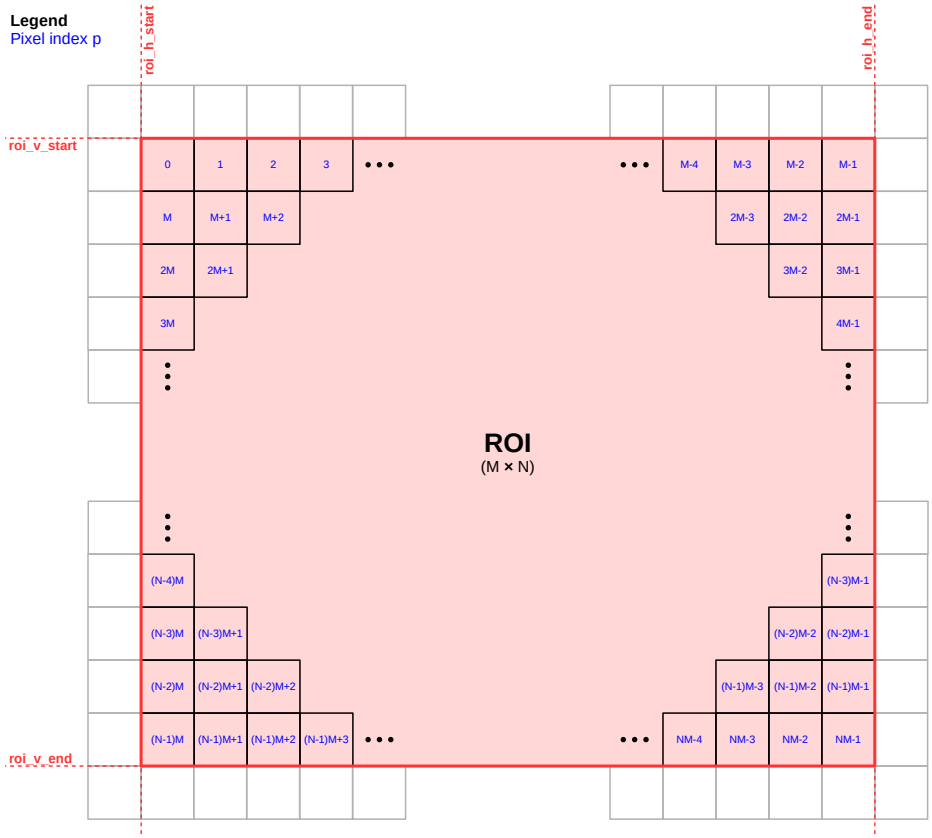

Figure 33: Pixel indexing based on ROI for test pattern computation

8.2.1. Full scale

The full scale test pattern consists of vertical stripes alternating between minimum and maximum values. Independent of the data format, the following 12 bit values are obtained:

- Even columns: 0x000
- Odd columns: 0xFFF

8.2.2. Ramp

The ramp test pattern consists of horizontally increasing gradients (following the pixel index). The values increase until the 12 bit limit is reached. Then, the gradient starts again. Additionally, a frame index increasing with every frame (single DCS) is added.

$$z = (p + f) \bmod 2^{12}$$, where p and f are pixel and frame indices, respectively.

8.2.3. PN9

The PN9 test pattern is computed by adding up powers of the pixel index. Additionally, a frame index increasing with every frame (single DCS) is added. Since the data format is limited to 12 bit, the higher order bits of the computed values are ignored.

$$z = (p^9 + p^5 + f) \bmod 2^{12}$$, where p and f are pixel and frame indices, respectively.

8.3. ADC testing

When disabling the global ADC bias current (**ibias_adc_ramp_gen_global_en** = '0'), the functionality of the ADC counters can be checked by selecting a predefined number of counting steps using register field **adc_counter_num_clock_cycles_sel** (register 0x7B, ANALOG_TEST_CTRL) and recording an image. The result is supposed to be a homogeneous image with the values specified in the register description. Note that this can also be checked on-chip using the range checker feature.

Furthermore, if the global ADC bias current is enabled (**ibias_adc_ramp_gen_global_en** = '1') and the integration time is set to zero (no integration), the received unsigned image data is supposed to be around 2048 (±100). If that is the case, this means that the whole ADCs (not only counters) are operating correctly. Again, this can also be checked on-chip using the range checker feature.

8.4. Detection of pixels stuck to vertical neighbor

Independent operation of vertically neighboring pixels can be tested by acquiring two (grayscale) images with the dual integration time feature (see Chapter 7.5.5):

- 1. frame: long integration time for even rows, short integration time for odd rows
- 2. frame: short integration time for even rows, long integration time for odd rows

Note: Checking for pixels stuck at a fixed value is also covered by this test.

8.5. Readout sequencer instruction parity checking

Every time the readout sequencer loads a new program line from the program memory, the validity of the loaded instruction is checked based on parity bits in the program code. If the check fails, the error flag **readout_sequencer_input_error** is set (see register ERROR_STATUS). The error status is also transmitted as part of the MIPI metadata (see Chapter 13.2.2.1).

8.6. MIPI error correction code for packet header

The MIPI CSI-2 protocol layer specifies an ECC code to be transmitted as part of every packet header (see Chapter 13.2) which is also implemented in the MIPI transmitter of the epc670. This allows the receiver side to correct a 1-bit error and detect 2-bit errors. Standard MIPI receiver hardware handles this automatically.

8.7. MIPI cyclic redundancy checksum for payload data

The MIPI CSI-2 protocol layer specifies a CRC code to be transmitted in the footer of every long packet which is also implemented in the MIPI transmitter of the epc670. The CRC is calculated according to the MIPI standard (see Chapter 13.2.2 for more details). Standard MIPI receiver hardware validates the CRC checksum automatically.

9. Temperature sensors

The characteristics of the four temperature sensors located next to the corners of the pixel-field are described in Table 13.

Table 13: Temperature sensor characteristics

Symbol	Description	Min.	Typ.	Max.	Unit
TMR_{TS}	Temperature measurement range, full specification	-40		+125 [4]	°C
		~600		~7300	LSB
$OFFSET_{TS,x}$	Temperature sensor offset for sensor x [1]	3469 [3]	3850 [3]	4231 [3]	LSB
$GAIN_{TS,x}$	Temperature sensor gain for sensor x [2]	39.7 [3]	46.0 [3]	52.4 [3]	LSB/°C
$SCALE_{OFFSET}$	Temperature sensor offset scale	-	3.0 [3]	-	-
$SCALE_{GAIN}$	Temperature sensor gain scale	-	0.05 [3]	-	-
R_{TS}	Resolution of the temperature sensor	-	13	-	bit
N_{TS}	Noise of temperature sensor output codes		2	4	LSB
LIN_{TS}	Linearity of temperature sensor over the full temperature range		±2	±4	°C
f_{TS}	Update rate of the temperature sensor (configurable)	0.1	1	10	Hz

[1] Uncalibrated temperature is measured at $\mathbf{T_{REF} = 27\,°C}$ per chip. Corresponding $OFFSET_{TS,x}$ is calculated and stored in EEPROM backed-up TEMP_SENSOR_x_CAL_OFFSET register during production test (wafer-level test).

[2] Temperature gain will be measured at $\mathbf{T_{REF2} = 85\,°C}$ over several lots/batches during characterization. Corresponding mean $GAIN_{TS,x}$ is calculated and stored in EEPROM backed-up TEMP_SENSOR_x_CAL_GAIN register during production test (wafer-level test).

[3] Value will be updated after chip characterization.

[4] When the maximum value of 8191 (ADC overflow) is read out from TEMP_SENSOR_x_hi/lo registers, this corresponds to a chip temperature ≥ 120 °C. When chip temperature drops below 120 °C, registers provide correct values again.

Absolute or precise relative temperature can be calculated using the values read from TEMP_SENSOR_x, TEMP_SENSOR_x_CAL_OFFSET and TEMP_SENSOR_x_CAL_GAIN (x ∈ {0, 1}) registers according to the following formula:

$$T_{CHIP,x} = \frac{\text{TEMP\_SENSOR\_x} - \text{OFFSET}_{TS,x}}{\text{GAIN}_{TS,x}} + T_{REF} \ ,$$

where

$$\text{OFFSET}_{TS,x}\,[\text{LSB}] = 3'850\ \text{LSB} + \text{SCALE}_{OFFSET} \cdot (\text{TEMP\_SENSOR\_x\_CAL\_OFFSET} - 127)\ \text{LSB} \ ,$$

$$\text{GAIN}_{TS,x}\,[\text{LSB/ºC}] = 46\ \text{LSB/ºC} + \text{SCALE}_{GAIN} \cdot (\text{TEMP\_SENSOR\_x\_CAL\_GAIN} - 127)\ \text{LSB/ºC} \ .$$

Example

Using T_{REF}, $SCALE_{OFFSET}$, $SCALE_{GAIN}$ from Table 13, let register values be:

- TEMP_SENSOR_x = 5'178 LSB (13 bit)
- TEMP_SENSOR_x_CAL_OFFSET = 74 LSB (8 bit)
- TEMP_SENSOR_x_CAL_GAIN = 216 LSB (8 bit)

Then, T_{CHIP} is calculated as:

- $OFFSET_{TS,x}$ = 3'850 LSB + 3.0 × (TEMP_SENSOR_x_CAL_OFFSET – 127) LSB = 3'691 LSB
- $GAIN_{TS,x}$ = 46 LSB/°C + 0.05 × (TEMP_SENSOR_x_CAL_GAIN – 127) LSB/°C = 50.45 LSB/°C
- $T_{CHIP,x}$ = (TEMP_SENSOR_x – $OFFSET_{TS,x}$) / $GAIN_{TS,x}$ + 27 °C = **56.47 °C**

10. Configuration

The epc670 control registers (RAM) are used for controlling all features of the chip. They are organized as 256 × 8 bit into 0x00 ... 0xFF address locations. The address space 0x80 ... 0xFF is EEPROM backed-up. EEPROM parameters in this section are stored permanently between the power off/on cycles. All registers can be accessed through I²C interface by the application CPU (see Chapter 12). Multiple byte registers are stored in big-endian format (byte with highest order at lowest address).

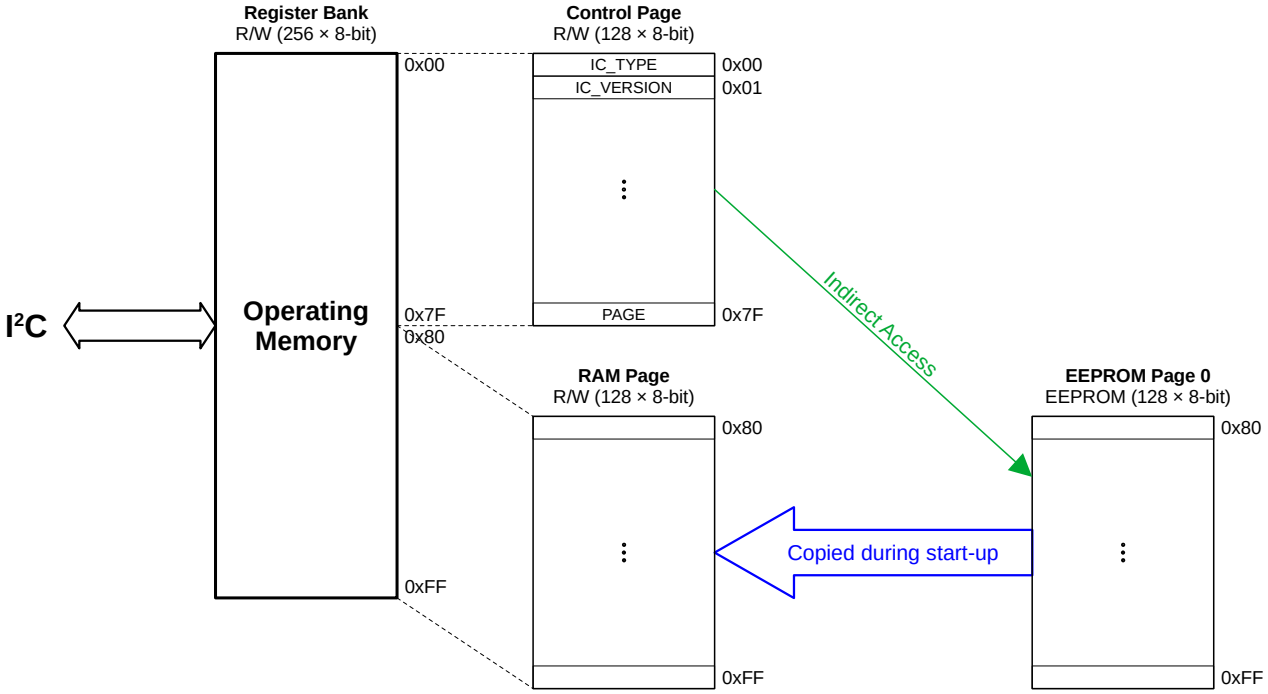

Figure 34: Memory map

10.1. Control page

The control page contains R/W accessible registers with default values during startup. The content can be changed via the I²C interface. The changed values are preserved as long as the IC is powered. They are set back to their default values with a reset.

10.2. RAM page

The RAM page contains R/W accessible registers with EEPROM copied values after startup. The content can be changed via the I²C interface. The changed values are preserved as long as the IC is powered. They are set back to EEPROM values with a reset.

10.3. EEPROM pages

An embedded 128 × 8-bit EEPROM (page 0) stores operation parameters as well as factory set trimming and calibration values. Data stored in page 0 is loaded into the RAM page during normal start-up.

Additionally, there are four extra EEPROM pages 1 to 4 (each 128 × 8-bit) to store the ADC gain calibration data.

10.4. Frame configuration table

epc670's frame configuration table (FCT) can be used to configure up to 16 frames (15 when using software shutter) of a DCS burst in advance. Updated FCT settings are used from the next shutter on (DCS burst) after any update. The following settings can be pre-configured:

- Dual MGX modes
  - Different demodulation phases for even and odd rows
  - Separate integration times for even and odd rows
- Modulation overrides (MGx on or off etc.)
- Distance offset
- Modulation clock divider
- Region of interest (ROI)
- Readout mode (ABS on or off etc.)
- Pixel binning mode
- Resolution reduction mode
- Delay line setting

For details, please refer to the descriptions of registers FCT_DATA_* (addresses 0x22 to 0x31). FCT access is controlled over register FCT_ACCESS_CTRL (0x32). For settings which are not taken from the FCT by default, the FCT has to be selected as the configuration source (see registers FCT_CFG_SEL and DISTANCE_OFFSET).

The configuration time is mainly determined by the I2C clock frequency (and the number of bytes to be written). The most efficient way to configure is to read back the entry to be modified first and only write to the FCT_DATA_* registers which shall be modified (see examples below).

Example – dual phase mode

The following configuration sequence enables dual phase mode with a DCS burst length of two frames:

1. Enable dual phase mode: MODULATION_CTRL = 0x88
2. Set number of frames per shutter to 2: SHUTTER_CTRL = 0x10
3. Load FCT entry 0: FCT_ACCESS_CTRL = 0x80
4. Select phases DCS0 and DCS1 for first frame: FCT_DATA_15 = 0x40
5. Save FCT entry 0: FCT_ACCESS_CTRL = 0xC0
6. Load FCT entry 1: FCT_ACCESS_CTRL = 0x81
7. Select phases DCS2 and DCS3 for second frame: FCT_DATA_15 = 0xE0
8. Save FCT entry 1: FCT_ACCESS_CTRL = 0xC1

Example – dual integration time mode

The following configuration sequence enables dual integration time mode for four DCSs:

1. Enable dual integration time mode: MODULATION_CTRL = 0x8C
2. Select integration times with registers INTEG_LEN_* and INTEG_LEN_DUAL*.
3. Load FCT entry 0: FCT_ACCESS_CTRL = 0x80
4. Select phases DCS0 and DCS0 for first frame: FCT_DATA_15 = 0x00
5. Save FCT entry 0: FCT_ACCESS_CTRL = 0xC0
6. Load FCT entry 1: FCT_ACCESS_CTRL = 0x81
7. Select phases DCS1 and DCS1 for second frame: FCT_DATA_15 = 0x50
8. Save FCT entry 1: FCT_ACCESS_CTRL = 0xC1
9. Load FCT entry 2: FCT_ACCESS_CTRL = 0x82
10. Select phases DCS2 and DCS2 for third frame: FCT_DATA_15 = 0xA0
11. Save FCT entry 2: FCT_ACCESS_CTRL = 0xC2
12. Load FCT entry 3: FCT_ACCESS_CTRL = 0x83
13. Select phases DCS3 and DCS3 for fourth frame: FCT_DATA_15 = 0xF0
14. Save FCT entry 3: FCT_ACCESS_CTRL = 0xC3

10.5. Readout sequencer

The readout of the pixel field including analog-to-digital conversion is controlled by the readout sequencer. It consists of a processing unit and a program memory. In order to optimize epc670's performance, the latest default sequencer program shall be loaded every time the chip is started up.

10.5.1. Pixel sequencer code write procedure

1. Start up the epc670 chip (power up and release reset).
2. Wait until the chip is started up and the sequencer is enabled (**readout_seq_en** = '1').
3. Write the pixel sequencer program code from Chapter 10.5.2 to the memory.

Important notes:

- This procedure has to be executed after every power up or after a chip reset release
- Never modify this code sequence. Otherwise, malfunction occurs.

10.5.2. Pixel sequencer program

```
# Default readout sequencer program – version 3
# The following sequence of I2C commands re-programs the readout sequencer to be on most actual func-
tionality.
#
# The syntax of the I2C commands to the imager is as follows:
# Writing: i2c w REGISTER_ADDRESS [RAM_ADDRESS DATA5 DATA4 DATA3 DATA2 DATA1 DATA0 ACCESS_CTRL]

i2c w 8F 00
i2c w 3A 10
```

```
i2c w 33 00 00 F0 00 01 E0 07 13
i2c w 33 01 00 C4 80 01 E0 07 13
i2c w 33 02 00 30 00 0D E0 07 13
i2c w 33 03 00 54 C6 01 E0 07 13
i2c w 33 04 00 A5 03 41 E0 07 13
i2c w 33 05 00 B5 45 81 E0 07 13
i2c w 33 06 00 33 00 01 E0 07 13
i2c w 33 07 00 F0 C6 85 E0 07 13
i2c w 33 08 00 01 FE 81 E0 07 13
i2c w 33 09 00 62 40 01 E0 07 13
i2c w 33 0A 00 13 80 01 E0 07 13
i2c w 33 0B 00 74 40 51 E0 07 13
i2c w 33 0C 00 40 80 01 E0 87 13
i2c w 33 0D 00 F0 03 89 E0 07 13
i2c w 33 0E 00 D0 00 03 E0 07 13
i2c w 33 0F 00 85 80 01 E0 07 13
i2c w 33 10 00 D0 C7 01 E0 07 13
i2c w 33 11 00 35 45 41 E0 07 13
i2c w 33 12 00 80 00 41 E0 07 13
i2c w 33 13 00 D0 C7 01 E0 07 13
i2c w 33 14 00 F0 00 01 E0 07 13
i2c w 33 15 00 D0 80 01 E0 07 13
i2c w 33 16 00 00 03 09 E0 07 13
i2c w 33 17 00 B0 81 81 E0 07 13
i2c w 33 18 00 70 03 49 E0 07 13
i2c w 33 19 00 B0 81 81 E0 07 13
i2c w 33 1A 00 D0 00 03 E0 07 13
i2c w 33 1B 00 85 80 01 E0 07 13
i2c w 33 1C 00 95 C0 01 E0 07 13
i2c w 33 1D 00 00 00 21 E0 07 13
i2c w 33 1E 00 F6 08 01 E0 07 13
i2c w 33 1F 00 F0 A2 01 E0 07 13
i2c w 33 20 00 16 48 81 E0 07 13
i2c w 33 21 00 80 9C 01 E0 07 13
i2c w 33 22 00 46 89 01 E0 07 13
i2c w 33 23 00 F0 96 01 E0 07 13
i2c w 33 24 00 A0 8A 01 E0 07 13
i2c w 33 25 00 C0 40 C0 A0 00 13
i2c w 33 26 00 20 40 40 20 04 13
i2c w 33 27 00 B1 80 00 00 04 13
i2c w 33 28 00 B0 00 05 E0 07 13
i2c w 33 29 00 10 40 81 E0 07 13
i2c w 33 2A 00 90 00 01 E0 01 13
i2c w 33 2B 00 B2 C0 01 E0 81 13
i2c w 33 2C 00 C1 09 40 E0 00 13
i2c w 33 2D 00 80 00 00 08 04 13
i2c w 33 2E 00 50 40 80 18 24 13
i2c w 33 2F 00 80 00 00 18 34 13
i2c w 33 30 00 60 00 00 18 3C 13
i2c w 33 31 00 02 00 00 18 34 13
i2c w 33 32 00 91 25 00 08 34 13
i2c w 33 33 00 56 E5 C0 20 74 13
i2c w 33 34 00 47 10 40 20 04 13
i2c w 33 35 00 17 C0 01 E0 07 13
i2c w 33 36 00 90 00 01 E0 01 13
i2c w 33 37 00 00 00 01 E0 81 13
i2c w 33 38 00 91 09 40 E0 20 13
i2c w 33 39 00 E0 00 00 04 04 13
i2c w 33 3A 00 30 40 80 14 24 13
i2c w 33 3B 00 E0 00 00 14 34 13
i2c w 33 3C 00 00 00 00 14 3C 13
i2c w 33 3D 00 62 00 00 14 34 13
i2c w 33 3E 00 F1 25 00 04 34 13
i2c w 33 3F 00 37 65 C0 20 74 13
i2c w 33 40 00 40 00 00 20 04 13
i2c w 33 41 00 17 C0 01 E0 07 13
i2c w 33 42 00 90 00 01 E0 01 13
i2c w 33 43 00 00 00 01 E0 81 13
i2c w 33 44 00 91 09 40 E0 20 13
i2c w 33 45 00 10 00 00 02 04 13
i2c w 33 46 00 C0 40 80 12 24 13
i2c w 33 47 00 10 00 00 12 34 13
i2c w 33 48 00 F0 00 00 12 3C 13
i2c w 33 49 00 92 00 00 12 34 13
i2c w 33 4A 00 01 25 00 02 34 13
i2c w 33 4B 00 27 25 C0 20 74 13
```

```
i2c w 33 4C 00 40 00 00 20 04 13
i2c w 33 4D 00 17 C0 01 E0 07 13
i2c w 33 4E 00 90 00 01 E0 01 13
i2c w 33 4F 00 00 00 01 E0 81 13
i2c w 33 50 00 91 09 40 E0 20 13
i2c w 33 51 00 70 00 00 01 04 13
i2c w 33 52 00 A0 40 80 11 24 13
i2c w 33 53 00 70 00 00 11 34 13
i2c w 33 54 00 90 00 00 11 3C 13
i2c w 33 55 00 F2 00 00 11 34 13
i2c w 33 56 00 61 25 00 01 34 13
i2c w 33 57 00 D0 A5 C0 20 74 13
i2c w 33 58 00 D0 00 05 E0 37 13
i2c w 33 59 00 70 40 81 E0 37 13
i2c w 33 5A 00 90 00 01 E0 01 13
i2c w 33 5B 00 B2 C0 01 E0 81 13
i2c w 33 5C 00 91 09 40 E0 20 13
i2c w 33 5D 00 40 00 00 0A 04 13
i2c w 33 5E 00 90 40 80 1A 24 13
i2c w 33 5F 00 40 00 00 1A 34 13
i2c w 33 60 00 A0 00 00 1A 3C 13
i2c w 33 61 00 C2 00 00 1A 34 13
i2c w 33 62 00 51 25 00 0A 34 13
i2c w 33 63 00 27 25 C0 20 74 13
i2c w 33 64 00 40 00 00 20 04 13
i2c w 33 65 00 17 C0 01 E0 07 13
i2c w 33 66 00 90 00 01 E0 01 13
i2c w 33 67 00 00 00 01 E0 81 13
i2c w 33 68 00 91 09 40 E0 20 13
i2c w 33 69 00 40 00 00 05 04 13
i2c w 33 6A 00 90 40 80 15 24 13
i2c w 33 6B 00 40 00 00 15 34 13
i2c w 33 6C 00 A0 00 00 15 3C 13
i2c w 33 6D 00 C2 00 00 15 34 13
i2c w 33 6E 00 51 25 00 05 34 13
i2c w 33 6F 00 D0 A5 C0 20 74 13
i2c w 33 70 00 D0 00 05 E0 37 13
i2c w 33 71 00 70 40 81 E0 37 13
i2c w 33 72 00 90 00 01 E0 01 13
i2c w 33 73 00 C2 C0 01 E0 80 13
i2c w 33 74 00 91 09 40 E0 20 13
i2c w 33 75 00 B0 00 00 0C 04 13
i2c w 33 76 00 60 40 80 1C 24 13
i2c w 33 77 00 B0 00 00 1C 34 13
i2c w 33 78 00 50 00 00 1C 3C 13
i2c w 33 79 00 32 00 00 1C 34 13
i2c w 33 7A 00 A1 25 00 0C 34 13
i2c w 33 7B 00 37 65 C0 20 74 13
i2c w 33 7C 00 40 00 00 20 04 13
i2c w 33 7D 00 17 C0 01 E0 07 13
i2c w 33 7E 00 90 00 01 E0 01 13
i2c w 33 7F 00 00 00 01 E0 81 13
i2c w 33 80 00 91 09 40 E0 20 13
i2c w 33 81 00 B0 00 00 03 04 13
i2c w 33 82 00 60 40 80 13 24 13
i2c w 33 83 00 B0 00 00 13 34 13
i2c w 33 84 00 50 00 00 13 3C 13
i2c w 33 85 00 32 00 00 13 34 13
i2c w 33 86 00 A1 25 00 03 34 13
i2c w 33 87 00 D0 A5 C0 20 74 13
i2c w 33 88 00 D0 00 05 E0 37 13
i2c w 33 89 00 70 40 81 E0 37 13
i2c w 33 8A 00 90 00 01 E0 01 13
i2c w 33 8B 00 B2 C0 01 E0 81 13
i2c w 33 8C 00 91 09 40 E0 20 13
i2c w 33 8D 00 D0 00 00 0F 04 13
i2c w 33 8E 00 00 40 80 1F 24 13
i2c w 33 8F 00 D0 00 00 1F 34 13
i2c w 33 90 00 30 00 00 1F 3C 13
i2c w 33 91 00 52 00 00 1F 34 13
i2c w 33 92 00 C1 25 00 0F 34 13
i2c w 33 93 00 D0 A5 C0 20 74 13
i2c w 33 94 00 47 80 00 00 34 13
i2c w 33 95 00 E2 80 00 20 74 13
i2c w 33 96 00 21 80 00 20 74 13
i2c w 33 97 00 40 00 00 20 04 13
```

```
i2c w 33 98 00 17 C0 01 E0 07 13
i2c w 33 99 00 C4 00 5D E0 01 13
i2c w 33 9A 00 00 00 01 E0 81 13
i2c w 33 9B 00 91 09 40 E0 20 13
i2c w 33 9C 00 80 00 00 08 04 13
i2c w 33 9D 00 50 40 80 18 24 13
i2c w 33 9E 00 80 00 00 18 34 13
i2c w 33 9F 00 60 00 00 18 3C 13
i2c w 33 A0 00 02 00 00 18 34 13
i2c w 33 A1 00 91 25 00 08 34 13
i2c w 33 A2 00 40 00 00 20 74 13
i2c w 33 A3 00 40 00 00 20 04 13
i2c w 33 A4 00 70 00 05 E0 04 13
i2c w 33 A5 00 D7 C0 01 E0 04 13
i2c w 33 A6 00 D8 00 01 E0 07 13
i2c w 33 A7 00 E1 40 05 E0 07 13
i2c w 3A 00
i2c w 8F 01
```

10.5.3. Pixel sequencer program reading and checking

It is possible to read the sequencer code back from the sequencer memory. This is useful to ensure that the sequencer code is correctly stored and was not accidentally changed during operation. The most efficient way is to trigger a sequencer RAM dump (see register FST_CTRL). The dump is then transmitted as a MIPI frame (see Chapter 13.2.2.4).

Alternatively, the program lines can be read back one by one over the register bank:

I²C command to imager	**Description / comment**
`shutter_ctrl = i2c r 90`	`# Store setting of register SHUTTER_CTRL`
`i2c w 90 00`	`# Disable acquisition (also logic level on SHUTTER pin shall be '0')`
`i2c w 8F 00`	`# Stop sequencer`
`i2c w 3A 10`	`# Enable sequencer RAM access`

`i2c w 33 00`
`i2c w 3A 11`
`data4 = i2c r 35`
`data3 = i2c r 36`
`data2 = i2c r 37`
`data1 = i2c r 38`
`data0 = i2c r 39`
	`# Set sequencer RAM address to be read (starting address 0x00)`
	`# Load sequencer instruction to register bank (read access)`
	`# No need to read SR_DATA_5 (not used)`

`i2c w 33 01`
`i2c w 3A 11`
`data4 = i2c r 35`
`data3 = i2c r 36`
`data2 = i2c r 37`
`data1 = i2c r 38`
`data0 = i2c r 39`
	`# Set sequencer RAM address to be read (e.g. address 0x01)`
	`# Load sequencer instruction to register bank (read access)`
	`# No need to read SR_DATA_5 (not used)`

`...`	`...`

`i2c w 33 A7`
`i2c w 3A 11`
`data4 = i2c r 35`
`data3 = i2c r 36`
`data2 = i2c r 37`
`data1 = i2c r 38`
`data0 = i2c r 39`
	`# Set sequencer RAM address to be read (last used address is 0xA7)`
	`# Load sequencer instruction to register bank (read access)`
	`# No need to read SR_DATA_5 (not used)`

`i2c w 3A 00`	`# Disable sequencer RAM access`
`i2c w 8F 01`	`# Start sequencer`
`i2c w 90 shutter_ctrl`	`# Restore setting of register SHUTTER_CTRL`

The following data should be received (compare to write commands in Chapter 10.5.2):

SR_ADDRESS	SR_DATA_4 (data4)	SR_DATA_3 (data3)	SR_DATA_2 (data2)	SR_DATA_1 (data1)	SR_DATA_0 (data0)
0x00	0xF0	0x00	0x01	0xE0	0x07
0x01	0xC4	0x80	0x01	0xE0	0x07
0x02	0x30	0x00	0x0D	0xE0	0x07
...	...	...	...	...	...
0xA7	0xE1	0x40	0x05	0xE0	0x07
0xA8	0x00	0x00	0x00	0x00	0x00

11. Register map

11.1. Abbreviations

Table 14 lists the abbreviations used for the register descriptions in the following sections.

Table 14: Overview of abbreviations used in register map

Short name	Long name	Description
R	Readable	Register value can be read.
W	Writable	Register value can be overwritten.
P	Protected	Register value can not be overwritten if protection is enabled.
C	Clearing	Register field is automatically cleared.
B	Bypassed	Bypass the register bank during normal readout. These bits can be written normally, but the readout value is set by the chip itself.
x	Don't care	
n/a	Not applicable	
'0'	Logic low level	
'1'	Logic high level	
MSB	Most significant byte	
LSB	Least significant byte	
msb	Most significant bit	
lsb	Least significant bit	
_hi	High byte (MSB) of a multi-byte register	
_mi	Middle byte of a multi-byte register	
_lo	Low byte (LSB) of a multi-byte register	
*	Shadowed registers	Shadowed (double-buffered) registers which can be updated on-the-fly any time, e.g. during a frame acquisition. The new values are copied and used at the start of the next DCS burst (shutter event).

11.2. Control page (0x00 ~ 0x7F)

The control page described in Table 15 is not EEPROM backed-up, i.e. default values can not be stored in the EEPROM.

Table 15: Control page

Addr.	Register	Type	Def.	Description
0x00	IC_TYPE	R	0x00	Device family identification
0x01	IC_VERSION	R	0x08	Version of epc670
0x02	reserved			
...	...			
0x20	reserved			
0x21	FST_CTRL	RW	0x00	**Functional Safety Test Control** **b7**: **fst_trigger [C]** starts an FST operation according to the **fst_mode_sel** setting when set: '**0**' = trigger processed '1' = triggered (automatically cleared) **b6..b3**: reserved, all '**0**', read-only **b2..b0**: **fst_mode_sel** defines the FST operation to be executed when **fst_trigger** is set. Not listed values are reserved. '**000**' = FST register dump '001' = FST readout sequencer RAM dump '100' = FST test pattern full scale '101' = FST test pattern ramp '110' = FST test pattern PN9 Note 1: FST trigger will be delayed internally until the end of an ongoing cwTOF DCS burst. Otherwise, it will start immediately. **fst_trigger** will be automatically cleared at the end of an FST frame. Note 2: Pulses on the SHUTTER pin are ignored during an FST frame.
0x22	FCT_DATA_15	RW	0x00	**Frame Configuration Table Entry Byte 15** **b7..b6**: **fct_dcs_num_mgx1** selects the DCS number (phase shift) for the outputs of modulator 1 (mga1, mgb1): '**00**' = DCS 0 (0 degrees) '01' = DCS 1 (90 degrees) '10' = DCS 2 (180 degrees) '11' = DCS 3 (270 degrees) **b5..b4**: **fct_dcs_num_mgx0** selects DCS number (phase shift) for the outputs of modulator 0 (mga0, mgb0): '**00**' = DCS 0 (0 degrees) '01' = DCS 1 (90 degrees) '10' = DCS 2 (180 degrees) '11' = DCS 3 (270 degrees) **b3..b2**: **fct_cds_gain_sel** selects CDS gain: '**00**' = 0.5 '01' = 1.0 '10' = 2.0 '11' = 4.0 **b1**: **fct_dual_mgx_mode_en** selects whether Dual MGX Mode is enabled: '**0**' = single modulator for all rows '1' = dual: modulator 0 for upper rows, modulator 1 for lower rows **b0**: reserved

Addr.	Register	Type	Def.	Description
0x23	FCT_DATA_14	RW	0x00	**Frame Configuration Table Entry Byte 14** **b7..b6**: reserved **b5**: **fct_override_integ_led_off** can be used to force the LED off: '**0**' = LED is modulated '1' = LED stays off entire integration time **b4**: **fct_override_integ_led_on** can be used to force the LED on over the entire integration time: '**0**' = LED is modulated '1' = LED stays on entire integration time (static) **b3**: **fct_override_integ_mgb_on** controls whether MGB stays on (conducting) over the entire integration time: '**0**' = MGB is modulated '1' = MGB stays on entire integration time **b2**: **fct_override_integ_mga_on** controls whether MGA stays on (conducting) over the entire integration time: '**0**' = MGA is modulated '1' = MGA stays on entire integration time **b1**: **fct_override_integ_mgb_off** controls whether MGB stays off (non-conducting) over the entire integration time: '**0**' = MGB is modulated '1' = MGB stays off entire integration time **b0**: **fct_override_integ_mga_off** controls whether MGA stays off (non-conducting) over the entire integration time: '**0**' = MGA is modulated '1' = MGA stays off entire integration time Note 1: **b0** has priority over **b2**, **b1** over **b3**, **b4** over **b5.** Note 2: **b5..b0** are ignored during LED preheat. Note 3: LED overrides are ignored for LED pins which are forced on (see LED_CTRL.**led_*_force_on**).
0x24	FCT_DATA_13	RW	0x00	**Frame Configuration Table Entry Byte 13** **b7..b1**: **fct_distance_offset** defines the delay as a number of main modulation clock cycles $T_{clk\_mod\_main}$ for the modulator outputs abd, abg, mga, mgb with respect to the led signal. The applied delay causes a phase shift which results in a distance shift. **b0**: reserved, def. '**0**', read-only
0x25	FCT_DATA_12	RW	0x00	**Frame Configuration Table Entry Byte 12** **b7**: reserved **b6..b2**: **fct_clk_div_mod** configures the clock divider which generates the modulation clock clk_mod. The frequency of the latter is derived by integer division of the frequency of the externally provided modulation clock: $f_{clk\_mod} = f_{clk\_mod\_ext}$ / (**clk_div_mod** + 1) **b1..b0**: **fct_roi_h_start_hi** is the high order bits of **fct_roi_h_start** which defines the index of the leftmost column of the region of interest. **fct_roi_h_start** has to be an integer multiple of four. In case of an invalid value, the value is internally corrected to the next lower valid value. Note: In case of column reduction/binning, the ROI width has to be an integer multiple of eight.
0x26	FCT_DATA_11	RW	0x00	**Frame Configuration Table Entry Byte 11** **b7**..**b1**: **fct_roi_h_start_lo** is the lowest bit of **fct_roi_h_start**. See description for **fct_roi_h_start_hi**. **b0**: **fct_roi_h_end_hi** is the high order bits of **fct_roi_h_end** which defines the index of the rightmost column of the region of interest. **fct_roi_h_end** has to be an integer multiple of four minus one. In case of column reduction/binning, the ROI width has to be an integer multiple of eight. Invalid values are internally corrected to the next higher valid value.
0x27	FCT_DATA_10	RW	0x00	**Frame Configuration Table Entry Byte 10** **b7..b0**: **fct_roi_h_end_lo** is the lowest two bits of **fct_roi_h_end**. See description for **fct_roi_h_end_hi**.

Addr.	Register	Type	Def.	Description
0x28	FCT_DATA_9	RW	0x00	**Frame Configuration Table Entry Byte 9** **b7..b3: fct_roi_v_start** defines the index of the topmost row of the region of interest. In order to keep the size of the frame configuration table reasonable, **fct_roi_v_start** s derived from the absolute row index as follows: **fct_roi_v_start** = roi_v_start / 8 **b2**..**b0: fct_roi_v_end_hi** is the high order bits of **fct_roi_v_end** which defines the index of the bottommost row of the region of interest. In order to keep the size of the frame configuration table reasonable, **fct_roi_v_end** is derived from the absolute row index as follows: **fct_roi_v_end** = (roi_v_end + 1) / 8 – 1
0x29	FCT_DATA_8	RW	0x00	**Frame Configuration Table Entry Byte 8** **b7..b6**: **fct_roi_v_end_lo** is the low nibble of **fct_roi_v_end**. See description for **fct_roi_v_end_hi**. **b5..b4: fct_readout_mode** defines the mode (ABS/TXG) for readout channels A and B: '00' = differential readout with ABS disabled '01' = single-ended readout from storage gate A '10' = single-ended readout from storage gate B '**11**' = differential readout with ABS enabled **b3..b2**: **fct_pixel_binning_mode** selects the spatial pixel binning mode (see Note 1): '**00**' = no binning '01' = horizontal binning '10' = vertical binning '11' = horizontal + vertical binning **b1..b0**: **fct_resolution_reduction_mode** selects the resolution reduction mode (see Note 2): '**00**' = full resolution '01' = half horizontal resolution (only every second column) '10' = half vertical resolution (only every second row) '11' = half horizontal and vertical resolution Note 1: When **dual_mgx_mode** = '1', allowed modes for **pixel_binning_mode** are '00' or '01', i.e. no binning or horizontal binning only. Note 2: When **pixel_binning_mode** != '00', **resolution_reduction_mode** is internally overwritten accordingly.
0x2A	FCT_DATA_7	RW	0x00	**Frame Configuration Table Entry Byte 7** **b7..b2**: **fct_led_delay_line_ctrl_coarse** controls the coarse part of the delay line for the LED control signal. The max. value for **fct_led_delay_line_ctrl_coarse** is 49. **b1**..**b0**: reserved
0x2B	FCT_DATA_6	RW	0x00	**Frame Configuration Table Entry Byte 6** **b7..b0**: reserved
0x2C	FCT_DATA_5	RW	0x00	**Frame Configuration Table Entry Byte 5** **b7..b0: fct_integ_len_hi** is the high byte of **fct_integ_len** which sets the length of the integration time as a multiple of the modulation clock period T_{clk_mod}: t_{integ} = **fct_integ_len** × T_{clk_mod} Note 1: For sine mode (cwTOF), **fct_nteg_len** must be an integer multiple of four. This ensures identical integration times for storage gates A and B. Note 2: In dual integration time mode (HDR mode), **fct_integ_len** controls the integration time of the upper rows within pixel row pairs. Note 3: If **fct_integ_len** = 0, integration phase is skipped.
0x2D	FCT_DATA_4	RW	0x00	**Frame Configuration Table Entry Byte 4** **b7..b0: fct_integ_len_mi** is the middle byte of **fct_integ_len**. See description of **fct_integ_len_hi**.

Addr.	Register	Type	Def.	Description
0x2E	FCT_DATA_3	RW	0x00	**Frame Configuration Table Entry Byte 3** **b7..b0: fct_integ_len_lo** is the low byte of **fct_integ_len**. See description of **fct_integ_len_hi**.
0x2F	FCT_DATA_2	RW	0x00	**Frame Configuration Table Entry Byte 2** **b7..b0: fct_integ_len_dual_hi** is the high byte of **fct_integ_len_dual** which sets the length of the integration time as a multiple of the modulation clock period T_{clk_mod} for the lower row within pixel row pairs in dual integration time mode (HDR mode): t_{integ_dual} = **fct_integ_len_dual** × T_{clk_mod} Note 1: For sine mode (cwTOF), **fct_integ_len_dual** must be an integer multiple of four. This ensures identical integration times for storage gates A and B. Note 2: When **integ_len_dual_en** = '0', **fct_integ_len_dual** is ignored and **integ_len** and **fct_integ_len** are used for modulator 1 when **cfg_sel_integ_len** = '0' and **cfg_sel_integ_len** = '1', respectively. Note 3: If **integ_len_dual** = 0 and **fct_integ_len** ≠ 0, modulation gates are not opened for lower rows within pixel row pairs. If **fct_integ_len_dual** ≠ 0 and **fct_integ_len** = 0, modulation gates are not opened for upper rows within pixel row pairs. Note 4: If **cfg_sel_integ_len_dual** = '1', **integ_len_dual_en** is ignored, **fct_integ_len_dual** is used for modulator 1.
0x30	FCT_DATA_1	RW	0x00	**Frame Configuration Table Entry Byte 1** **b7..b0: fct_integ_len_dual_mi** is the middle byte of **fct_integ_len_dual**. See description of **fct_integ_len_dual_hi**.
0x31	FCT_DATA_0	RW	0x00	**Frame Configuration Table Entry Byte 0** **b7..b0: fct_integ_len_dual_lo** is the low byte of **fct_integ_len_dual**. See description of **fct_integ_len_dual_hi**.
0x32	FCT_ACCESS_CTRL	RW	0x00	**Frame Configuration Table Control** **b7**: **fct_access_trigger [C]** triggers frame configuration table access: '**0**' = trigger processed '1' = triggered (automatically cleared) **b6**: **fct_access_mode_sel** selects frame configuration table access mode: '**0**' = read: Load data from FCT at **fct_addr** into FCT_DATA_* registers. '1' = write: Copy data from FCT_DATA_* registers into FCT at **fct_addr**. **b5..b4**: reserved, def. '**0**', read-only **b3..b0**: **fct_addr** (def. **0**) sets the address for the frame configuration table access.
0x33	SR_ADDRESS	RW	0x00	**b7..b0**: **seq_ram_addr** (def. **0**) defines the address for accessing a sequencer RAM based on the settings of SR_ACCESS_CTRL.
0x34	reserved	R	0x00	reserved
0x35	SR_DATA_4	RW	0x00	**b7..b0**: **seq_ram_data_4** (def. **0**) is byte 4 ([39:32]) of the data written to or read back from a sequencer RAM based on the settings of SR_ADDRESS and SR_ACCESS_CTRL.
0x36	SR_DATA_3	RW	0x00	**b7..b0**: **seq_ram_data_3** (def. **0**) is byte 3 ([31:24]) of the data written to or read back from a sequencer RAM based on the settings of SR_ADDRESS and SR_ACCESS_CTRL.
0x37	SR_DATA_2	RW	0x00	**b7..b0**: **seq_ram_data_2** (def. **0**) is byte 2 ([23:16]) of the data written to or read back from a sequencer RAM based on the settings of SR_ADDRESS and SR_ACCESS_CTRL.
0x38	SR_DATA_1	RW	0x00	**b7..b0**: **seq_ram_data_1** (def. **0**) is byte 1 ([15:8]) of the data written to or read back from a sequencer RAM based on the settings of SR_ADDRESS and SR_ACCESS_CTRL.
0x39	SR_DATA_0	RW	0x00	**b7..b0**: **seq_ram_data_0** (def. **0**) is byte 0 ([7:0]) of the data written to or read back from a sequencer RAM based on the settings of SR_ADDRESS and SR_ACCESS_CTRL.

Addr.	Register	Type	Def.	Description
0x3A	SR_ACCESS_CTRL	RW	0x00	**Sequencer RAM Access Control** **b7..b5**: reserved, all '**0**', read-only **b4**: **seq_ram_access_en_readout** enables the access to the readout sequencer RAM: '**0**' = access disabled '1' = access enabled **b3..b2**: reserved, all '**0**', read-only **b1**: **seq_ram_access_mode_sel** selects sequencer RAM access mode: '**0**' = read '1' = write **b0**: **seq_ram_access_trigger [C]** triggers sequencer RAM access: '**0**' = trigger processed '1' = triggered (automatically cleared)
0x3B	ADC_GAIN_CAL_CTRL	RW	0x00	**b7..b2**: reserved **b1**: **adc_gain_cal_done [R]** reports the status of the ADC gain calibration: '**0**' = ADC gain uncalibrated / calibration in progress '1' = ADC gain calibrated **b0**: reserved
0x3C	reserved			
...	...			
0x3E	reserved			
0x3F	ERROR_STATUS	RW	0x00	**Error Status Flags** **b7**: **error_status_clear [C]** can be used to clear any asserted error status flags: '**0**' = no clear request pending '1' = clear request pending (automatically cleared) **b6..b1**: reserved, all '**0**', read-only **b0**: **readout_sequencer_input_error [R]** is asserted when the parity check of the readout sequencer input from the sequencer RAM is failing: '**0**' = no error '1' = error
0x40	reserved			
…	…			
0x55	reserved			
0x56	READOUT_RANGE_MAX_hi	W	0x00	**b15..b12:** reserved **b11..b0: readout_range_max** (def. **0**) defines the upper limit of the range used by the frame checker to evaluate the read out pixel data, write-only.
0x57	READOUT_RANGE_MAX_lo	W	0x00	
0x58	READOUT_RANGE_MIN_hi	W	0x00	**b15..b12:** reserved **b11..b0: readout_range_min** (def. **0**) defines the lower limit of the range used by the frame checker to evaluate the read out pixel data, write-only.
0x59	READOUT_RANGE_MIN_lo	W	0x00	
0x5A	RANGE_FAILURES_OVER_OR_MID_hi	RW	0x00	**b15..b12:** reserved **b12..b0: range_failures_over_or_mid** reports or defines the number of failures above and in between the given threshold and thresholds, respectively: Note 1: *Read:* Number of over- or in-range failures (def. **0**) *Write:* Allowed number of over- or in-range failures (internal def. **0**) Note 2: Readout only works if the frame checker is enabled FRAME_CHECKER_CTRL.**enable_frame_checker**. Note 3: Always write and read the *hi* register before the *lo* register.
0x5B	RANGE_FAILURES_OVER_OR_MID_lo	RW	0x00	
0x5C	RANGE_FAILURES_UNDER_hi	RW	0x00	**b15..b13:** reserved, all '**0**', read-only

Addr.	Register	Type	Def.	Description
0x5D	RANGE_FAILURES_UNDER_lo	RW	0x00	**b12..b0: range_failures_under** reports or defines the number of failures below the given threshold. Note 1: *Read:* Number of under-range failures (def. **0**) *Write:* Allowed number of under-range failures (internal def. **0**) Note 2: Readout only works if the frame checker is enabled FRAME_CHECKER_CTRL.**enable_frame_checker**. Note 3: Always write and read the *hi* register before the *lo* register.
0x5E	FRAME_CHECKER_CTRL	RW	0x00	**Frame Checker Control** **b7**: **in_range_check** selects the checking mode: **'0'** = Check if values are outside the defined range. '1' = Check if values are inside the defined range. **b6**: **reset_frame_checker [C]** resets the range checker results. **'0'** = normal '1' = reset (automatically cleared) **b5**: **enable_frame_checker** controls whether range checker is enabled (see Note 1): **'0'** = disable '1' = enable **b4..b2**: reserved, all '**0**', read-only **b1: bypass_failure_fifo** controls whether the FIFO for range failures are bypassed (use the register value for all frames). **'0'** = normal '1' = bypass **b0**: reserved, def. '**0**', read-only Note: Any readout from the range checker status and range failures registers does only work if the frame checker is enabled.
0x5F	FRAME_CHECKER_STATUS	R	0x02	**Frame Checker Status** **b7..b2**: reserved, all '**0**', read-only **b1**: **frame_checker_valid** indicates whether frame_checker_fail flag is valid: '0' = invalid '**1**' = valid **b0: frame_checker_fail** indicates failures during range check: '**0**' = no failure '1' = failure Note: Readout only works if the frame checker is enabled. See FRAME_CHECKER_CTRL.**enable_frame_checker**.
0x60	reserved			
…	…			
0x65	reserved			
0x66	TEMP_SENSOR_0_hi	R	0x00	**Temperature Sensor 0 Data** **b15..b13:** reserved, all '0', read-only **b12..b0: temp_sensor_0** provides data from temperature sensor 0.
0x67	TEMP_SENSOR_0_lo	R	0x00	
0x68	TEMP_SENSOR_1_hi	R	0x00	**Temperature Sensor 1 Data** **b15..b13:** reserved, all '0', read-only **b12..b0: temp_sensor_1** provides data from temperature sensor 1.
0x69	TEMP_SENSOR_1_lo	R	0x00	
0x6A	TEMP_SENSOR_2_hi	R	0x00	**Temperature Sensor 2 Data** **b15..b13:** reserved, all '0', read-only **b12..b0: temp_sensor_2** provides data from temperature sensor 2.
0x6B	TEMP_SENSOR_2_lo	R	0x00	
0x6C	TEMP_SENSOR_3_hi	R	0x00	**Temperature Sensor 3 Data** **b15..b13:** reserved, all '0', read-only **b12..b0: temp_sensor_3** provides data from temperature sensor 3.
0x6D	TEMP_SENSOR_3_lo	R	0x00	

Addr.	Register	Type	Def.	Description
0x6E	TEMP_SENSOR_CTRL	RW	0x01	**b7..b2**: reserved, all '**0**' **b1..b0**: **sel_temp_sensor_rate** selects the temperature measurement rate for all temperature sensors assuming f_{clk_temp} = 2 MHz as follows (see Note): '00' = 10 Hz '**01**' = 1 Hz '10' = 0.1 Hz '11' = reserved (10 Hz) Note: Measurement rate change becomes effective only after the temperature sensor has been turned off and on again (see register POWER_CTRL_STATIC_1).
0x6F	LED_DELAY_CTRL_COARSE*	RW	0x00	**b7..b6**: reserved, all '**0**', read-only **b5..b0**: **led_delay_line_ctrl_coarse** (def. **0**) configures the delay of the coarse part of the delay line for the LED control signal. **led_delay_line_ctrl_coarse** must not be set larger than 49.
0x70	reserved			
0x71	reserved			
0x72	LED_DELAY_COARSE_RB	R	0x00	**b7..b6**: reserved, all '**0**', read-only **b5..b0**: **led_delay_line_coarse_rb** provides the selection read back of the coarse part of the delay line for the LED control signal.
0x73	reserved			
…	…			
0x7A	reserved			
0x7B	ANALOG_TEST_CTRL	RW	0x83	**Analog Test Control** **b7**..**b3**: reserved, all '**0**', read-only **b2**..**b0**: **adc_counter_num_clock_cycles_sel** selects the global number of clock cycles for the ADC counters per AD conversion. Not listed values are reserved. '000' = 0 (ADC output unsigned: 0x000) '001' = 683 (ADC output unsigned: 0x555 if comparators do not trigger) '010' = 1366 (ADC output unsigned: 0xAAB if comp. do not trigger) '**011**' = 2048 (ADC output unsigned: 0xFFF if comp. do not trigger) '100' = 2049 (ADC output unsigned: 0x001 if comp. do not trigger)
0x7C	reserved			
...	...			
0x7F	reserved			

11.3. EEPROM page 0 (0x80 ~ 0xFF)

This page is EEPROM backed-up, i.e. the default values stored in the EEPROM are loaded during start-up.

Table 16: EEPROM page 0 (Registers 0x80 to 0xFF, EEPROM backed-up)

Addr.	Register	Type	Def.	Description
0x80	reserved			
...	...			
0x84	reserved			
0x85	CLK_DIVIDER_MOD*	RW	0x21	**b7:** reserved, def. '**0**', read-only **b6**: **clk_mod_sel** controls which clock source is used to generate the modulation clock: '**0**' = external clock from pad CLK_MOD '1' = internal clock from main PLL with frequency pre-divided by 2 **b5**: reserved, def. '**1**' **b4..b0**: **clk_div_mod** (def. **1**) configures the clock divider which generates the modulation clock clk_mod. The frequency of the latter is derived by integer division of the frequency of the externally provided modulation clock (optionally internally doubled): $f_{clk\_mod} = f_{clk\_mod\_ext}$ / (**clk_div_mod** + 1) = $f_{clk\_mod\_ext}$ / 2 = **40 MHz**
0x86	reserved			
...	...			
0x8E	reserved			
0x8F	SEQUENCER_CTRL	RW	0x01	**b7..b1**: reserved **b0**: **readout_seq_en** controls whether the readout sequencer is enabled (see Note 3): '**0**' = disabled '1' = enabled Note: **readout_seq_en** will automatically be set after start-up.
0x90	SHUTTER_CTRL	RW	0x30	**b7..b4**: **dcs_num_sel** (def. **3**) selects the number of differential correlation samples (DCS burst length) which are triggered per shutter: N_{DCS} = **dcs_num_sel** + 1 = **4** **b3..b2**: reserved, all '**0**', read-only **b1**: **multi_frame_en** controls whether setting **shutter** = '1' triggers a single or continuing (video mode) DCS bursts: '**0**' = single DCS burst '1' = continuing DCS bursts **b0**: **shutter [C]** triggers a DCS burst: '**0**' = shutter request processed '1' = shutter request pending (automatically cleared) Note 1: Multiple (continuous) frame acquisition can be enabled by setting **multi_frame_en** = '1' and **shutter** = '1' together. Note 2: **shutter** is not automatically cleared when **multi_frame_en** = '1'. Set back **shutter** = '0' to stop multiple (continuous) frame acquisition.

Addr.	Register	Type	Def.	Description
0x91	MODULATION_CTRL*	RW	0x80	**b7..b4**: reserved **b3**: **dual_mgx_mode_en** selects whether Dual MGX Mode is enabled: '**0**' = single modulator for all rows '1' = dual: modulator 0 for upper rows, modulator 1 for lower rows **b2: integ_len_dual_en** controls whether a separate integration time for modulator 1 is used (see Note 1): **'0'** = **integ_len** for both modulators '1' = **integ_len** and **integ_len_dual** for modulators 0 and 1, respectively **b1**: reserved **b0**: **mod_sel** selects the modulation/integration mode: **'0'** = cwTOF '1' = Grayscale (see Note 2) Note 1: **integ_len_dual_en** must be enabled only when **dual_mgx_mode** = '1', otherwise it is not effective. Note 2: Channel for grayscale mode and readout mode can be selected using override bits in frame configuration table.
0x92	DEMODULATION_DELAY	RW	0x01	**b7..b4**: reserved, all '**0**', read-only **b3..b0**: **mgx_del_sel** controls the delay in clk_mod_ext periods $T_{clk\_mod\_ext}$ on the abd, abg, mga, mgb signals in order to compensate internal LED driver delay: 0 = no delay **1** = 1 clock cycle 2 = 2 clock cycles … 12 = 12 clock cycles 13 = reserved (12 clocks) … 15 = reserved (12 clocks) Note: When **mgx_del_sel** set > 12 (i.e. 13, 14 or 15), the demodulation delay is internally limited to 12 $T_{clk\_mod\_main}$.
0x93	DISTANCE_OFFSET*	RW	0x00	**b7**: **cfg_sel_distance_offset** controls where the setting for the distance offset is coming from: '**0**' = **distance_offset** (register bank) '1' = **fct_distance_offset** (frame configuration table) **b6..b0**: **distance_offset** defines the delay as a number of main modulation clock cycles $T_{clk\_mod\_main}$ for the modulator outputs abd, abg, mga, mgb with respect to the led signal. The applied delay causes a phase shift which results in a distance shift.

Addr.	Register	Type	Def.	Description
0x94	FCT_CFG_SEL*	RW	0x00	**Frame Configuration Selection Control** **b7**: **cfg_sel_cds_gain_sel** controls where the setting for the CDS gain selection is coming from: '**0**' = **cds_gain_sel** (register bank) '1' = **fct_cds_gain_sel** (frame configuration table) **b6**: **cfg_sel_dual_mgx_mode_en** controls where the setting for the enabling of the Dual MGX Mode is coming from: '**0**' = **dual_mgx_mode_en** (register bank) '1' = **fct_dual_mgx_mode_en** (frame configuration table) **b5**: **cfg_sel_roi** controls where the ROI settings are coming from: '**0**' = **roi_h_start/end, roi_v_start/end** (register bank) '1' = **fct_ roi_h_start/end, fct_roi_v_start/end** (frame config. table) **b4**: **cfg_sel_pixel_binning_mode** controls where the setting for the spatial pixel binning is coming from: '**0**' = **pixel_binning_mode** (register bank) '1' = **fct_pixel_binning_mode** (frame configuration table) **b3**: **cfg_sel_resolution_reduction_mode** controls where the setting for the resolution reduction is coming from: '**0**' = **resolution_reduction_mode** (register bank) '1' = **fct_resolution_reduction_mode** (frame configuration table) **b2**: **cfg_sel_led_delay_ctrl** controls where the settings for the delay line to delay the LED control signal are coming from: '**0**' = **led_delay_line_ctrl_coarse** (register bank) '1' = **fct_led_delay_line_ctrl_coarse** (frame configuration table) **b1**: **cfg_sel_integ_len** controls where the setting for the integration time is coming from: '**0**' = **integ_len** (register bank) '1' = **fct_integ_len** (frame configuration table) **b0**: **cfg_sel_integ_len_dual** controls where the setting for the second integration for dual integration time mode is coming from: '**0**' = **integ_len_dual** (register bank) '1' = **fct_integ_len_dual** (frame configuration table)
0x95	LED_CTRL*	RW	0x01	**b7**: reserved **b6**: **led_delay_mux_sel** selects which LED control signal is delayed: '**0**' = LED 0 '1' = LED 1 **b5**: **led_1_inv_en** controls whether the polarity of the LED_1 control signal is inverted: '**0**' = normal '1' = inverted **b4**: **led_1_force_on** controls whether LED_1 is permanently on (torch function, independent of integration time, overriding modulator output): '**0**' = LED permanently off when not controlled by modulator '1' = LED permanently on **b3**: **led_1_en** controls whether LED_1 pin (pad driver) is enabled (see Note): '**0**' = disabled (LED_1 = '0', i.e. 0 V) '1' = enabled **b2**: **led_0_inv_en** controls whether the polarity of the LED_0 control signal is inverted: '**0**' = normal '1' = inverted **b1**: **led_0_force_on** controls whether LED_0 is permanently on (torch function, independent of integration time, overriding modulator output): '**0**' = LED permanently off when not controlled by modulator '1' = LED permanently on **b0**: **led_0_en** controls whether LED_0 pin (pad driver) is enabled (see Note): '0' = disabled (LED_0 = '0', i.e. 0 V) '**1**' = enabled Note: When LED pad output drivers are disabled, internal pull-downs are enabled.
0x96	ROI_H_START_hi*	RW	0x00	**b15..b9**: reserved, all '**0**', read-only

Addr.	Register	Type	Def.	Description
0x97	ROI_H_START_lo*	RW	0x04	**b8..b0**: **roi_h_start** (def. **4**) defines the index of the leftmost column of the region of interest. **roi_h_start** has to be an integer multiple of four. In case of an invalid value, the value is internally corrected to the next lower valid value. Note: In case of column reduction/binning, the ROI width has to be an integer multiple of eight.
0x98	ROI_H_END_hi*	RW	0x01	**b15..b9:** reserved, all '**0**', read-only
0x99	ROI_H_END_lo*	RW	0x43	**b8..b0**: **roi_h_end** (def. **323**) defines the index of the rightmost column of the region of interest. **roi_h_end** has to be an integer multiple of four minus one. In case of column reduction/binning, the ROI width has to be an integer multiple of eight. Invalid values are internally corrected to the next higher valid value.
0x9A	ROI_V_START*	RW	0x08	**b7..b0: roi_v_start** (def. **8**) defines the index of the topmost row of the region of interest. **roi_h_start** has to be an integer multiple of eight. In case of an invalid value, the value is internally corrected to the next lower valid value.
0x9B	ROI_V_END*	RW	0xF7	**b7..b0: roi_v_end** (def. **247**) defines the index of the bottommost row of the region of interest. **roi_v_end** has to be an integer multiple of eight minus one. In case of an invalid value, the value is internally corrected to the next higher valid value.
0x9C	LED_PREHEAT_LEN_hi*	RW	0x00	**b15..b0: led_preheat_len** (def. **160**) sets the length of the LED preheat time (before actual integration) as a multiple of the modulation clock period T_{clk_mod}: $t_{preheat}$ = **led_preheat_len** × T_{clk_mod} = **160** × T_{clk_mod} = 4 **µs** Note 1: For sine mode (cwTOF), **led_preheat_len** should be an integer multiple of four. Note 2: If **led_preheat_len** = 0, preheat phase is skipped.
0x9D	LED_PREHEAT_LEN_lo*	RW	0xA0	
0x9E	reserved			
0x9F	reserved			
0xA0	INTEG_LEN_hi*	RW	0x00	**b23..b0: integ_len** (def. **400**) sets the length of the integration time as a multiple of the modulation clock period T_{clk_mod}: t_{integ} = **integ_len** × T_{clk_mod} = **400** × T_{clk_mod} = **10 µs** Note 1: For sine mode (cwTOF), **integ_len** must be an integer multiple of four. This ensures identical integration times for storage gates A and B. Note 2: In dual integration time mode (HDR mode), **integ_len** controls the integration time of the upper rows within pixel row pairs. Note 3: If **integ_len** = 0, integration phase is skipped.
0xA1	INTEG_LEN_mi*	RW	0x01	
0xA2	INTEG_LEN_lo*	RW	0x90	
0xA3	INTEG_LEN_DUAL_hi*	RW	0x00	**b23..b0: integ_len_dual** (def. **400**) sets the length of the integration time as a multiple of the modulation clock period T_{clk_mod} for the lower row within pixel row pairs in dual integration time mode (HDR mode): t_{integ_dual} = **integ_len_dual** × T_{clk_mod} = **400** × T_{clk_mod} = **10 µs** Note 1: For sine mode (cwTOF), **integ_len_dual** must be an integer multiple of four. This ensures identical integration times for storage gates A and B. Note 2: When **integ_len_dual_en** = '1', the register value of **integ_len_dual** is ignored and **integ_len** is internally used for modulator 1. Note 3: If **integ_len_dual_en** = '1', **integ_len_dual** = 0 and **integ_len** ≠ 0, modulation gates are not opened for lower rows within pixel row pairs. If **integ_len_dual_en** = '1', **integ_len_dual** ≠ 0 and **integ_len** = 0, modulation gates are not opened for upper rows within pixel row pairs.
0xA4	INTEG_LEN_DUAL_mi*	RW	0x01	
0xA5	INTEG_LEN_DUAL_lo*	RW	0x90	
0xA6	reserved			

Addr.	Register	Type	Def.	Description
0xA7	READOUT_CTRL*	RW	0x10	**b7: col_readout_dir_sel** controls in which direction the columns per pixel field row are read out/transmitted: '**0**' = left to right (PCB design view; small to large column indices) '1' = right to left (PCB design view; large to small column indices) **b6: row_readout_dir_sel** controls in which direction the pixel field rows are read out/transmitted: '**0**' = top to bottom (small to large row indices) '1' = bottom to top (large to small row indices) **b5..b4:** reserved **b3..b2**: **pixel_binning_mode** selects the spatial pixel binning mode (see Note 1): '**00**' = no binning '01' = horizontal binning '10' = vertical binning '11' = horizontal + vertical binning **b1..b0**: **resolution_reduction_mode** selects the resolution reduction mode (see Notes 2 and 3): '**00**' = full resolution '01' = half horizontal resolution (only every second column) '10' = half vertical resolution (only every second row) '11' = half horizontal and vertical resolution Note 1: **dual_mgx_mode** has priority over **pixel_binning_mode**. When **dual_mgx_mode** = '1', allowed modes for **pixel_binning_mode** are '00' or '01', i.e. no binning or horizontal binning only. **pixel_binning_mode** gets internally corrected in case of invalid combinations. Note 2: **dual_mgx_mode** has priority over **resolution_reduction_mode**. When **dual_mgx_mode** = '1', allowed modes for **resolution_reduction_mode** are '00' or '01', i.e. no resolution reduction or horizontal resolution reduction only. **resolution_reduction_mode** gets internally corrected in case of invalid combinations. Note 3: When **pixel_binning_mode** ≠ '00', **resolution_reduction_mode** is internally overwritten accordingly.
0xA8	reserved			
...	...			
0xAA	reserved			
0xAB	CDS_ADC_CTRL*	RW	0x01	**b7..b4:** reserved, all '**0**' **b3..b0: cds_gain_sel** (def. **1**) determines the CDS gain according to the following formula: CDS gain = (**b3** · 217 + **b2** · 107 + **b1** · 53 + **b0** · 27 + 27) / 107 ≈ **0.5**
0xAC	reserved			
0xAD	reserved			
0xAE	RSIR_1	RW	0x01	**b7..b0**: **rsir_1** (def. **1**) sets the MIPI frame blanking time in steps of 25 µs. Note: **rsir_1** must be greater than zero.
0xAF	reserved			
...	...			
0xB2	reserved			

Addr.	Register	Type	Def.	Description
0xB3	MIPI_CTRL	RW	0x01	**MIPI Control** **b7**: **reset_frame_number [C]** resets the CSI-2 frame number in frame start and frame end packets to 1: '**0**' = no frame counter reset pending '1' = frame counter reset pending (automatically cleared) **b6**: **disable_data_sat** controls whether the image data is replaced with the reserved code for saturation (unsigned: 0x000, signed 0x800, see **mipi_data_signedness_sel**) when the saturation flag is set (see Note 2): '**0**' = enabled: image data is replaced with saturation code in case of pixel saturation '1' = disabled: raw image data independent of pixel saturation **b5**: **disable_line_sync** controls whether line synchronization packet (line start (LS) and line end (LE)) are enabled: '**0**' = LS and LE are surrounding long packet transmitting line data '1' = no LS and LE packets **b4**: **disable_embedded_data** controls whether metadata is transmitted at the beginning of a CSI-2 frame (Note 1): '**0**' = enabled (metadata before image data) '1' = disabled (only image data) **b3**: reserved, def. '**0**', read-only **b2**: **mipi_data_signedness_sel** selects the signedness for data transmitted in RAW12 format: '**0**' = unsigned (0 to 4095) '1' = signed (two's complement, -2048 to +2047) **b1..b0**: **mipi_clock_mode_sel** selects the clocking mode for data transmission over MIPI: '00' = clock enabled and disabled for every single packet '**01**' = clock enabled for row bursts '10' = clock enabled from start to end of a frame '11' = clock always running Note 1: FST register dump frames never have metadata. Note 2: **disable_data_sat** is independent of the value of **sat_window**. It only affects the merging of saturation flag and pixel data on the digital side, i.e. it is independent of the performance of the saturation detection circuitry. Note 3: Signed data format (**mipi_data_signedness_sel** = '1') is not supported for test pattern data.
0xB4	reserved			
0xB5	reserved			
0xB6	I2C_ADDRESS	RW	0x3C	**b7**: reserved, '0', read-only **b6..b0**: **i2c_dev_addr** (def. '**0111100**') sets the 7-bit I^2C device address and is programmable via direct access from I2C or from EEPROM during start-up, followed by an I2C general call 'reload'.
0xB7	reserved			
...	...			
0xE7	reserved			
0xE8	TEMP_SENSOR_0_CAL_OFFSET	RWP	0xFF	**Temperature Sensor 0 Offset Correction for SW** Range of -127 LSB to +127 LSB. 0x7F (127) corresponds to 0 LSB offset correction (def. **255**, meaning uncalibrated).
0xE9	TEMP_SENSOR_0_CAL_GAIN	RWP	0xFF	**Temperature Sensor 0 Gain Correction for SW** Range of -127 LSB/°C to +127 LSB/°C. 0x7F (127) corresponds to 0 LSB/°C gain correction (def. **255**, meaning uncalibrated).
0xEA	TEMP_SENSOR_1_CAL_OFFSET	RWP	0xFF	**Temperature Sensor 1 Offset Correction for SW** Range of -127 LSB to +127 LSB. 0x7F (127) corresponds to 0 LSB offset correction (def. **255**, meaning uncalibrated).
0xEB	TEMP_SENSOR_1_CAL_GAIN	RWP	0xFF	**Temperature Sensor 1 Gain Correction for SW** Range of -127 LSB/°C to +127 LSB/°C. 0x7F (127) corresponds to 0 LSB/°C gain correction (def. **255**, meaning uncalibrated).

Addr.	Register	Type	Def.	Description
0xEC	TEMP_SENSOR_2_CAL_OFFSET	RWP	0xFF	**Temperature Sensor 2 Offset Correction for SW** Range of -127 LSB to +127 LSB. 0x7F (127) corresponds to 0 LSB offset correction (def. **255**, meaning uncalibrated).
0xED	TEMP_SENSOR_2_CAL_GAIN	RWP	0xFF	**Temperature Sensor 2 Gain Correction for SW** Range of -127 LSB/°C to +127 LSB/°C. 0x7F (127) corresponds to 0 LSB/°C gain correction (def. **255**, meaning uncalibrated).
0xEE	TEMP_SENSOR_3_CAL_OFFSET	RWP	0xFF	**Temperature Sensor 3 Offset Correction for SW** Range of -127 LSB to +127 LSB. 0x7F (127) corresponds to 0 LSB offset correction (def. **255**, meaning uncalibrated).
0xEF	TEMP_SENSOR_3_CAL_GAIN	RWP	0xFF	**Temperature Sensor 3 Gain Correction for SW** Range of -127 LSB/°C to +127 LSB/°C. 0x7F (127) corresponds to 0 LSB/°C gain correction (def. **255**, meaning uncalibrated).
0xF0	reserved			
…	…			
0xF5	reserved			
0xF6	WAFER_ID_hi	RWP		Wafer ID **b15..b0**: **wafer_id** is a unique identification number over all epc670 wafers.
0xF7	WAFER_ID_lo	RWP		
0xF8	CHIP_ID_hi	RWP		Chip ID **b15..b0**: **chip_id** is a unique identification number per wafer.
0xF9	CHIP_ID_lo	RWP		
0xFA	PART_TYPE	RWP	0x0F	Part type
0xFB	PART_VERSION	RWP	0x08	Part version
0xFC	reserved			
…	…			
0xFF	reserved			

12. Inter-integrated circuit (I^2C) interface

12.1. Electrical interface

The I^2C bus interface allows accessing the RW registers and programming the EEPROM registers which store the configuration parameters. It is the only interface through which the configuration registers (Figure 34) can be accessed by the application. The epc670 operates as a slave device according to the I^2C specification with a transfer rate of up to 400 kbit/s in Fast Mode (FM) or 1 Mbit/s in Fast Mode plus (FM+). The I^2C master such as an external CPU can set the transfer speed simply by driving the SCL input at the desired frequency (up to 1 MHz). Therefore, no prior register configuration or setting is necessary.

I^2C specification is supported in epc670 with following remarks/exceptions:

- 7-bit addressing only is supported.
- Clock stretching is supported.
- General calls are supported:
  - Address 0x00 followed by 0x06: software reset of the whole chip
  - Address 0x00 followed by 0x04: device address reload
- Software reset is supported.
- Other uses of I^2C bus are not supported.

12.2. Device addressing

The epc670 7-bit I^2C device address is 011'1100 by default (see Figure 35). In a multi-camera application with epc670 devices connected on the same I^2C bus as slaves or together with other I^2C slaves talking to a single I^2C master, different device addresses can be preprogrammed in the EEPROM using the 5 MSBs in order to correctly identify different epc670 slaves on the bus. Additionally, the whole 7-bit address can be reconfigure using register I2C_ADDRESS (0xB6) in combination with a general call reload command.

msb							lsb
0	1	1	1	1	0	0	**R/$\overline{W}$**

Figure 35: Device address through I^2C

12.3. I^2C bus protocol notation

The following notation is used:

- S **S**TART condition
- P STO**P** condition
- A **A**cknowledge last byte (ACK)
- $\overline{A}$ **N**ot-Acknowledge last byte (NACK)
- Shaded part of protocol: transmitted by master
- Unshaded part of protocol: transmitted by epc670

12.4. I^2C bus timing

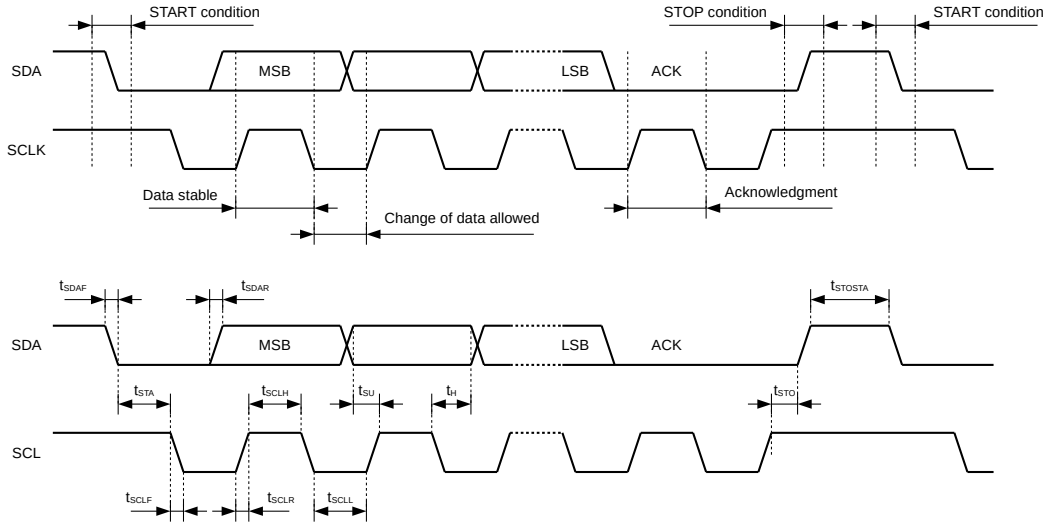

Figure 36: I^2C bus timing

Table 17: I²C bus timing: Timing parameters (FM+)

Symbol	Parameter	Min.	Max.	Units
t_{SCLL}	SCL clock low time	0.5		µs
t_{SCLH}	SCL clock high time	0.26		µs
t_{SU}	SDA setup time	50		ns
t_{H}	SDA hold time		0	ns
t_{SDAR} / t_{SCLR}	SDA and SCL rise time		120	ns
t_{SDAF} / t_{SCLF}	SDA and SCL fall time		120	ns
t_{STA}	Start condition hold time	0.26		µs
t_{STO}	Stop condition setup time	0.26		µs
t_{STOSTA}	Stop to start condition time (bus free)	0.5		µs
C_b	Capacitive load for each bus line (max. load, consider timing depending on the pull-up resistor and clock frequency)		550	pF

12.5. I²C commands

12.5.1. Software reset

(0x00, 0x06) issues a software reset which has the same effect like a hardware reset.

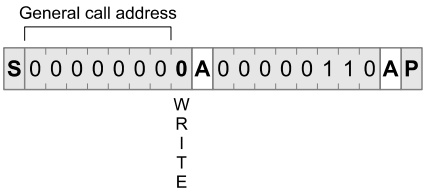

Figure 37: Software reset through I²C

12.5.2. Device address reload

(0x00, 0x04) activates the I²C address stored in register I2C_ADDRESS (0xB6).

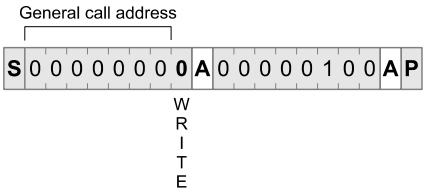

Figure 38: Device address reload through I²C

12.5.3. Write single-byte

During a single-byte write, only one register is written. After the device address is transmitted, the master has to transmit the register address and the write data in two I²C data packets (Figure 39). The access is terminated by a STOP condition.

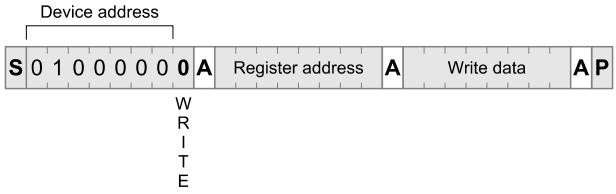

Figure 39: Single-byte Write access through I²C

12.5.4. Write multi-byte

During a multi-byte write operation, the master transmits the device address and the address of the first register to be written. All subsequent bytes until the STOP condition are interpreted as write data packets (Figure 40). The write address pointer is incremented internally. Do not transmit more data bytes than there are addresses from the provided start address till the limit of the address space (0xFF).

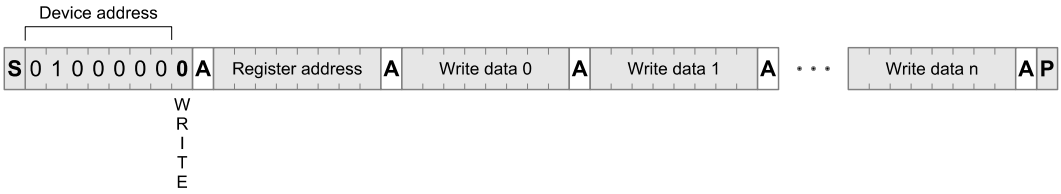

Figure 40: Multi-byte Write access through I²C

12.5.5. Read single-byte

The master transmits first the device address with a write command. Next, it writes the register address to be read. Then, the master transmits the device address again with a read command where the epc670 answers with the data stored in the addressed register. Finally, the master terminates the read sequence with a NACK and a STOP (Figure 41).

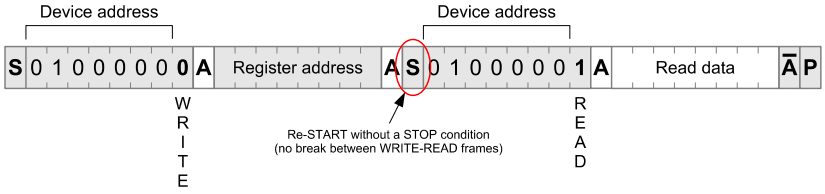

Figure 41: Single-byte Read access through I²C

12.5.6. Read multi-byte

The master transmits first the device address and the address of the first register to be read. After the epc670 is addressed with a read command, epc670 answers with read data bytes until the master does not acknowledge a byte. The master is expected to terminate the access with a STOP condition thereafter (Figure 42). During the access the read address pointer is incremented internally. Do not read more data bytes than there are addresses from the provided start address till the limit of the address space (0xFF).

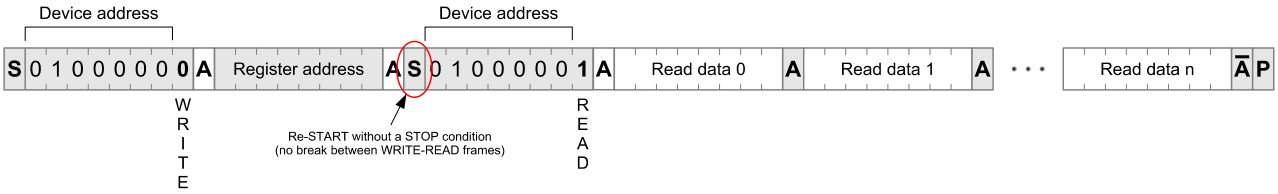

Figure 42: Multi-byte Read access through I²C

12.6. Command timing

The operating modes of the epc670 are initialized, activated, deactivated and monitored by sending several single or multi-byte write and read command sequences through I²C interface. This section lists and explains available commands together with their access time (f_{SCL} = 1 MHz → t_{SCL} = 1 µs).

There is no particular order defined for sending the commands. The only requirement is having no on-going frame acquisition process when updating non-shadowed registers. The registers marked with * in the register map can be updated on-the-fly during a frame acquisition. New values are used for the next DCS burst.

Table 18: I²C control command sequence examples

Command	Description	Length [Bytes]	Time [µs]
Single-byte write	Single-byte write to control registers	3	29
Multiple-byte write	Multiple-byte write (n bytes) to control registers	2 + n	20 + n × 9
Single-byte read	Single-byte read from control registers	4	39
Multiple-byte read	Multiple-byte read (n bytes) from control registers	3 + n	30 + n × 9
Number of frames set	The number of DCS frames to be recorded per shutter (**dcs_num_sel**)	3	29
Integration time set	Integration time setting (**integ_len**)	5	47
Dual Integration time set	Dual integ. time setting (**integ_len** and **integ_len_dual**)	8	74
ROI set	Region of interest configuration (**roi_h_start/end** and **roi_v_start/end**)	8	74
Shutter	Start frame acquisition by using the shutter control register	3	29
EEPROM indirect single write	Indirect single write operation to EEPROM	9	20 ms
EEPROM indirect single read	Indirect single read operation from EEPROM	10	97

13. Mobile industry processor interface (MIPI)

Image data is transmitted with the MIPI CSI-2 protocol over a MIPI D-PHY interface with four differential data lanes DATA_x_P/N plus a differential clock lane DATA_CLK_P/N. Figure 43 is depicting the MIPI CSI-2 transmitter on system level. The data transmission protocol MIPI CSI-2 is packet-based and described in Chapter 13.2. The MIPI D-PHY signal timing on the physical interface is described in Chapter 13.3.

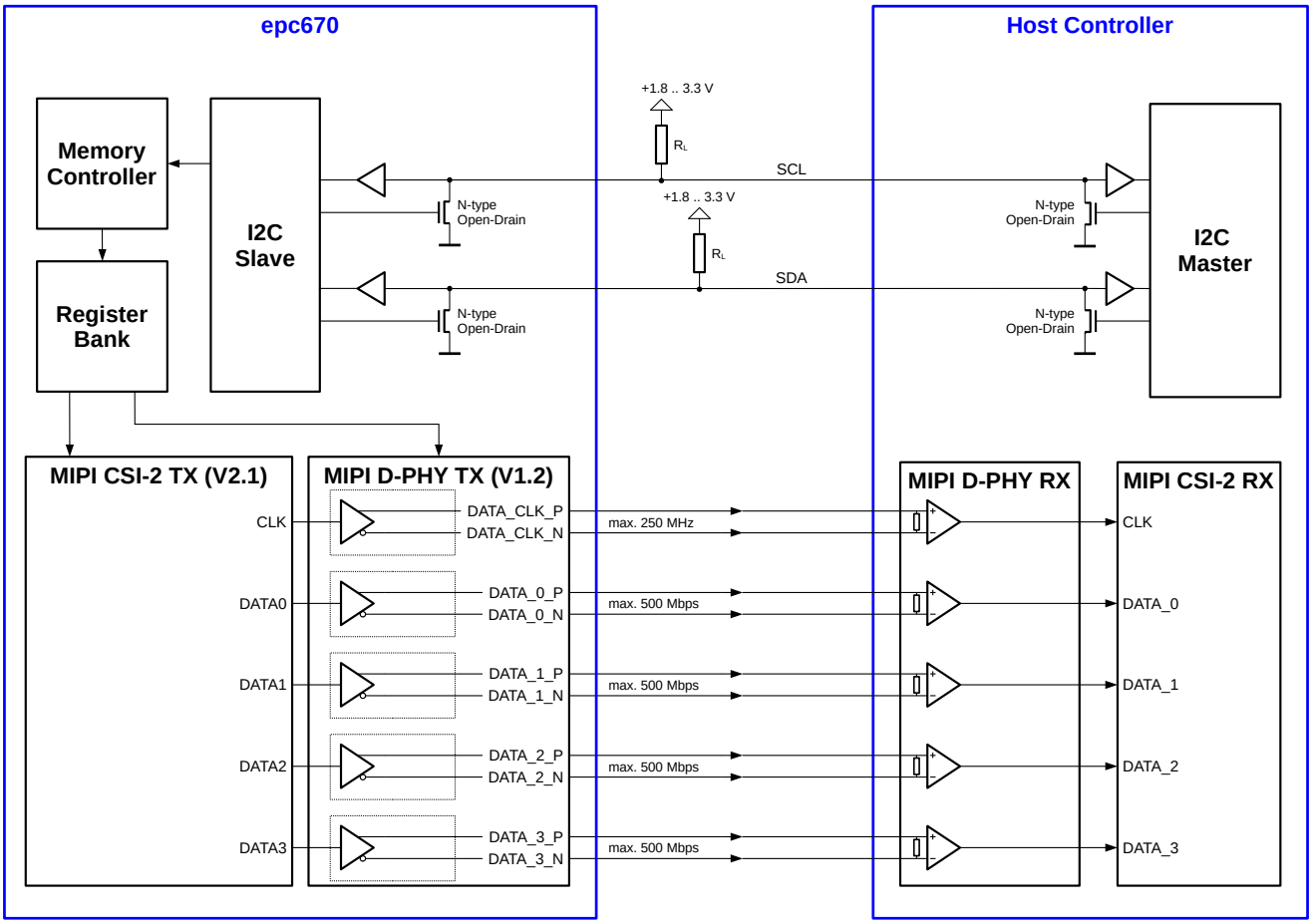

Figure 43: Connection of data and control interfaces to host controller

13.1. Key features

- D-PHY TX version: 1.2
  - Clock and data from epc670
  - Max. wire length to receiver: 100 mm
- CSI-2 master controller version: 2.1
  - Unidirectional data transmission from epc670 (master) to host controller (slave)
- Supported data formats:
  - RAW12 (dtype 0x2C) for image data
  - Generic 8-bit Long Packet Data Type (dtype 0x12) for metadata (embedded row data, no image data)
- Number of Lanes: 4 @ 500 Mbps DDR (2 Gbps in total in high-speed mode)
- Synchronization packets:
  - Frame start and frame end
  - Line start and line end
- Camera control interface: I^2C (3.3 V open-drain pads)
  - epc670 is an I^2C slave.

13.2. MIPI CSI-2 protocol

The MIPI CSI-2 data transmission protocol is packet-based. It is a generic protocol based on two main packet types, i.e. short and long packets. Each packet has a data identifier (DI) which defines its structure. Table 19 provides an overview of all packet types. The grayed out types are not relevant for epc670.

Table 19: Packet type overview

Data identifier range	Packet type	Packet subtype
0x00 – 0x07	Short packet	Synchronization packet
0x08 – 0x0F	Short packet	Generic short packet
0x10 – 0x17	Long packet	Generic 8-bit long packet
0x18 – 0x1F	Long packet	YUV image data
0x20 – 0x27	Long packet	RGB image data
0x28 – 0x2F	Long packet	RAW image data
0x30 – 0x37	Long packet	User defined byte-based data
0x38 – 0x3F	reserved	reserved

13.2.1. Short packet

A short packet is a packet with a length of only 4 bytes. It contains a 6-bit data identifier (DI), a 16-bit short packet data field and a 6-bit error correction code (ECC) as depicted in Figure 44. Virtual channels are not supported. VC and VCX bits are all zero therefore. A description of the fields is given in Table 20. Only synchronization short packets are transmitted by epc670 (see Table 19).

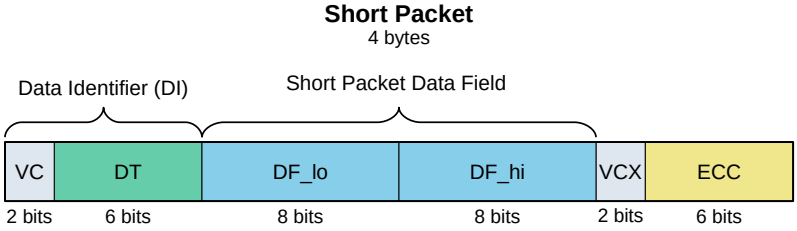

Figure 44: Structure of a short packet

Table 20: Overview of fields in a short packet

Acronym	Full name	Description
VC	**V**irtual **C**hannel low bits	Least significant 2 bits of 4-bit identifier for virtual channels. Not supported – all bits are zero.
DT	**D**ata **T**ype	Defines the function of the packet and the content of the data field.
DF_lo	**D**ata **F**ield low byte	16-bit payload of a short packet Low byte is transmitted before high byte.
DF_hi	**D**ata **F**ield high byte	
VCX	**V**irtual **C**hannel high bits	Most significant 2 bits of 4-bit identifier for virtual channels. Not supported – all bits are zero.
ECC	**E**rror **C**orrection **C**ode	Parity bits which allow single-bit errors across a short packet to be corrected, and two-bit errors to be detected.

Data transmission is organized in frames and lines where each frame consists of a least one line. A MIPI CSI-2 frame corresponds to a DCS, a grayscale image, a test pattern image, a register dump or a readout sequencer RAM dump. The number of lines directly corresponds to the number of rows selected in the ROI for images. Hence, the max. number of lines for epc670 is 328. For register and readout sequencer RAM dumps, there is only a single line per frame. Beginning and end of frames and lines are marked by dedicated short packets called synchronization packets. The different synchronization packets are listed in Table 21 and illustrated in Figure 45.

Table 21: Synchronization packets

Data type	Name	Acronym
0x00	Frame Start	FS
0x01	Frame End	FE
0x02	Line Start	LS
0x03	Line End	LE

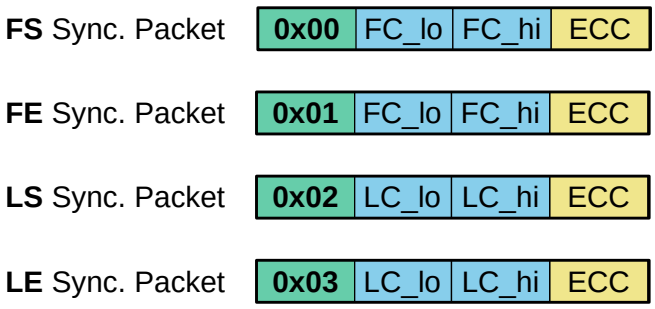

Figure 45: Synchronization packets

Frame count (FC) and line count (LC) are identical for FS-FE and LS-LE packet pairs:

- FS and FE: FC (FC_hi & FC_lo concatenated) is incremented for every MIPI frame in the range 1 to 256. When 256 is reached, the frame counter is set back to 1.
- LS and LE: LC (LC_hi & LC_lo concatenated) is incremented for every line per frame and per data type. LC is starting from 1 for every frame.

The error correction code (ECC) is calculated over the first three bytes plus VCX (see Figure 44, ignored in Figure 45).

13.2.2. Long Packet

A long packet is container for up to 65'535 bytes of data. The structure of a data packet is depicted in Figure 46. A description of the fields is given in Table 22.

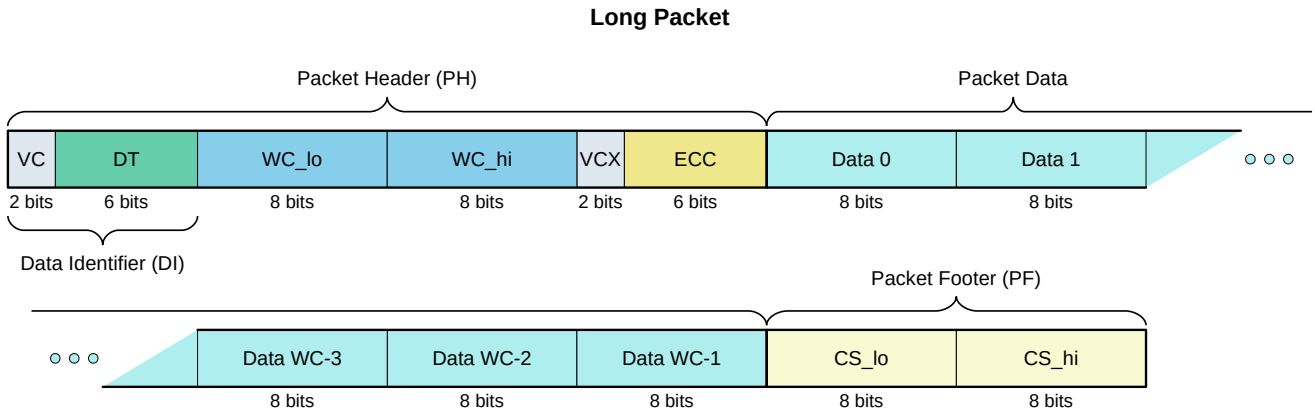

Figure 46: Structure of a long packet

Table 22: Overview of fields in a long packet

Byte Designator	Full Name	Description
PH	**P**acket **H**eader	Packet segment containing information for identification of packet
VC	**V**irtual **C**hannel low bits	Least significant 2 bits of 4-bit identifier for virtual channels. Not supported – all bits are zero.
DT	**D**ata **T**ype	6-bit code defining the function of the packet and the payload data format
WC_lo	**W**ord **C**ount low byte	16-bit packet data word count defining the length of the payload (number of bytes)
WC_hi	**W**ord **C**ount high byte	
VCX	**V**irtual **C**hannel high bits	Most significant 2 bits of 4-bit identifier for virtual channels. Not supported – all bits are zero.
ECC	**E**rror **C**orrection **C**ode	Parity bits which allow single-bit errors across the packet header to be corrected, and two-bit errors to be detected.
PD	**P**acket **D**ata	Packet segment containing actual payload data
Data *	Payload Data byte	Application-specific payload data (8 bit word width)
PF	**P**acket **F**ooter	Packet segment containing payload integrity information (checksum)
CS_lo	**P**ayload **C**hecksum **H**igh byte	16-bit CRC checksum which allows to check for transmission errors in the payload – the checksum is calculated based on the generator polynomial $x^{16}+x^{12}+x^5+x^0$ with a shift register initialization of 0xFFFF.
CS_hi	**P**ayload **C**hecksum **L**ow Byte	

Amount and structure of the payload are defined by the data type. Table 23 describes the payload content for all data packet types transmitted by the epc670.

Table 23: Payload depending on data type

Data type	Data type name	Usage	Payload [bit]
0x12	Embedded 8-bit non image data	Metadata	28 × 8 = 224
0x2C	RAW12 image data	Image data (pixel or test pattern)	(number of columns) × 12 Default ROI: 320 × 12 = 3’840
0x30	User defined 8-bit data type 1	Register dump	256 × 8 = 2'048
0x31	User defined 8-bit data type 2	Readout sequencer RAM dump	256 × 5 × 8 = 10’240

13.2.2.1 Metadata

Additionally to the image data, there is one line of metadata at the beginning of each frame with image data in RAW12 format (pixel data or test pattern). The metadata is transmitted in the embedded 8-bit non image data format. Figure 47 illustrates the distribution of the metadata over the 28 payload bytes. The content is listed in Table 24.

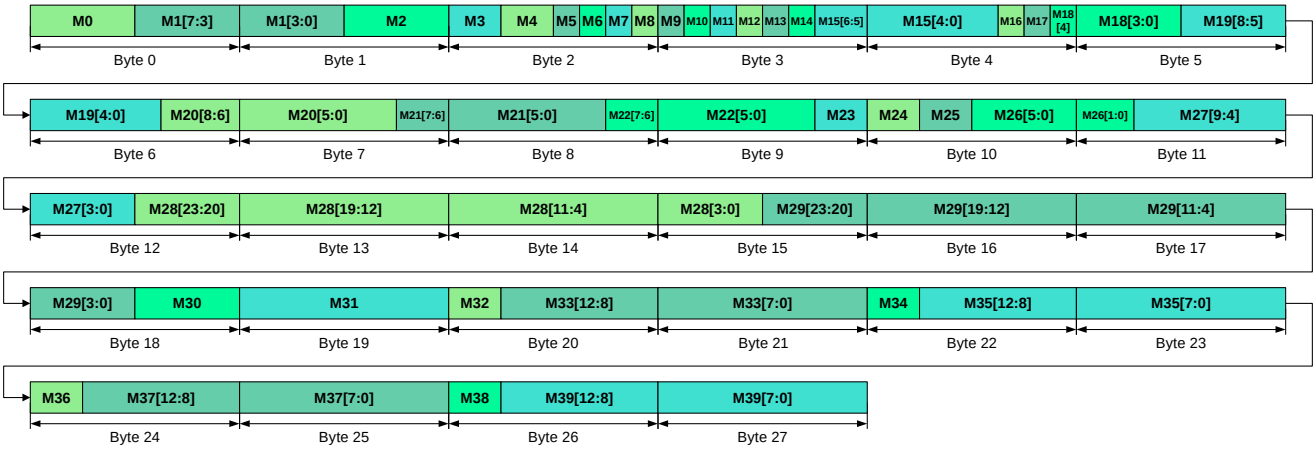

Figure 47: Data organization for metadata

Table 24: Metadata content

Designator	Metadata	Number of bits
M0	Frame type (see Table 25)	4
M1	Error status (see register ERROR_STATUS (0x3F))	8
M2	*reserved*	4
M3	**dcs_num_mgx1**	2
M4	**dcs_num_mgx0**	2
M5	**dual_mgx_mode_en**	1
M6	*reserved*	1
M7	*reserved*	1
M8	*reserved*	1
M9	**override_integ_led_off**	1
M10	**override_integ_led_on**	1
M11	**override_integ_mgb_on**	1
M12	**override_integ_mga_on**	1
M13	**override_integ_mgb_off**	1
M14	**override_integ_mga_off**	1
M15	**distance_offset**	7
M16	**clk_mod_sel**	1
M17	*reserved*	1
M18	**clk_div_mod**	5
M19	**roi_h_start**	9
M20	**roi_h_end**	9

Designator	Metadata	Number of bits
M21	**roi_v_start**	8
M22	**roi_v_end**	8
M23	**readout_mode**	2
M24	**pixel_binning_mode**	2
M25	**resolution_reduction_mode**	2
M26	**led_delay_line_ctrl_coarse**	6
M27	*reserved*	10
M28	**integ_len**	24
M29	**integ_len_dual**	24
M30	**cds_gain_sel**	4
M31	*reserved*	8
M32	*reserved*	3
M33	**temp_sensor_0**	13
M34	*reserved*	3
M35	**temp_sensor_1**	13
M36	*reserved*	3
M37	**temp_sensor_2**	13
M38	*reserved*	3
M39	**temp_sensor_3**	13

Table 25: Frame types

Frame type code	Frame type
0000	*reserved*
0001	cwTOF DCS
0010	Grayscale
0011	*reserved*
0100	*reserved*
0101	*reserved*
0110	*reserved*
0111	*reserved*
1000	Register dump
1001	Readout sequencer RAM dump
1010	*reserved*
1011	*reserved*
1100	FST test pattern full scale
1101	FST test pattern ramp
1110	FST test pattern PN9
1111	*reserved*

13.2.2.2 Image data

The payload data organization for image data of one pixel field row in RAW12 format is illustrated in Figure 48.

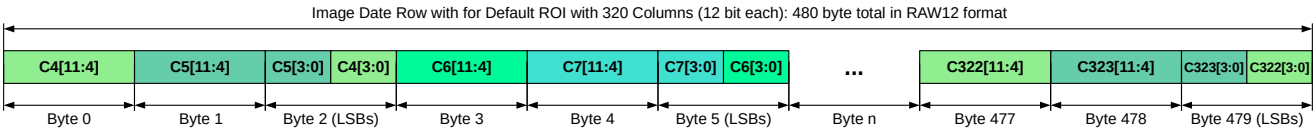

Figure 48: Data organization for RAW12 format

Table 26 explains the meaning of pixel data values. The signedness can be selected using register field **mipi_data_signedness_sel** in register MIPI_CTRL (0xB3).

Table 26: Pixel data ranges and signedness

Unsigned (default)		Signed		Validity	Description
Hexadecimal	Decimal	Hexadecimal	Decimal		
0x000	0	0x800	-2048	Invalid	Pixel saturation or ADC underflow [1]
0x001 to 0x7FF	1 to 2047	0x801 to 0xFFF	-2047 to -1	Valid	More charge in acquisition channel B than A (see Chapter 7.3)
0x800	2048	0x000	0	Valid	Same amount of charge in acquisition channels A and B
0x801 to 0xFFE	2049 to 4094	0x001 to 0x7FE	+1 to +2046	Valid	More charge in acquisition channel A than B (see Chapter 7.3)
0xFFF	4095	0x7FF	+2047	Invalid	ADC overflow

[1] If data saturation is enabled (MIPI_CTRL.**disable_data_sat** = '0'), it is not possible to distinguish pixel saturation and ADC underflow. However, the value is invalid in both cases.

13.2.2.3 Register dump

The payload data organization for a register dump is illustrated in Figure 49. Each register **R*** is one byte long.

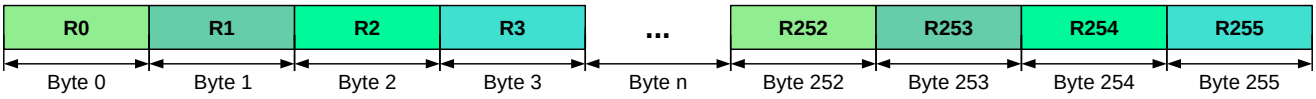

Figure 49: Data organization for user-defined data type 1

13.2.2.4 Readout sequencer RAM dump

The payload data organization for a readout sequencer RAM dump is illustrated in Figure 50. Each readout sequencer RAM line **S*** is five bytes long.

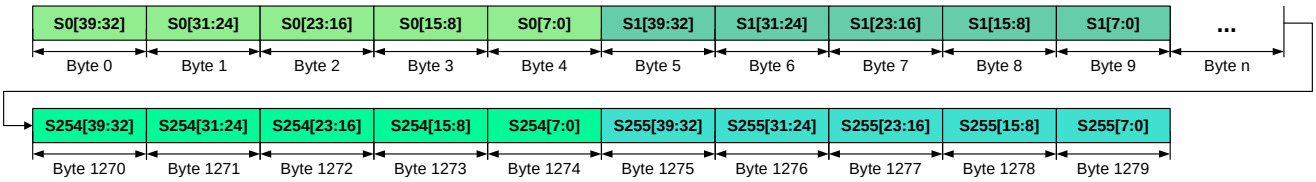

Figure 50: Data organization for user-defined data type 2

13.2.3. Frame transmission

Figure 51 depicts the MIPI CSI-2 image data frame structure for a DCS with default ROI. The frame is structured with the synchronization packets described in Chapter 13.2.1. Before the actual image data is transmitted, one line of metadata (embedded data) is transmitted. The length of the embedded data packet is fixed and independent of the number of columns selected by the ROI. Depending on the shape of the image to be transmitted, the number and length of the MIPI CSI-2 lines varies.

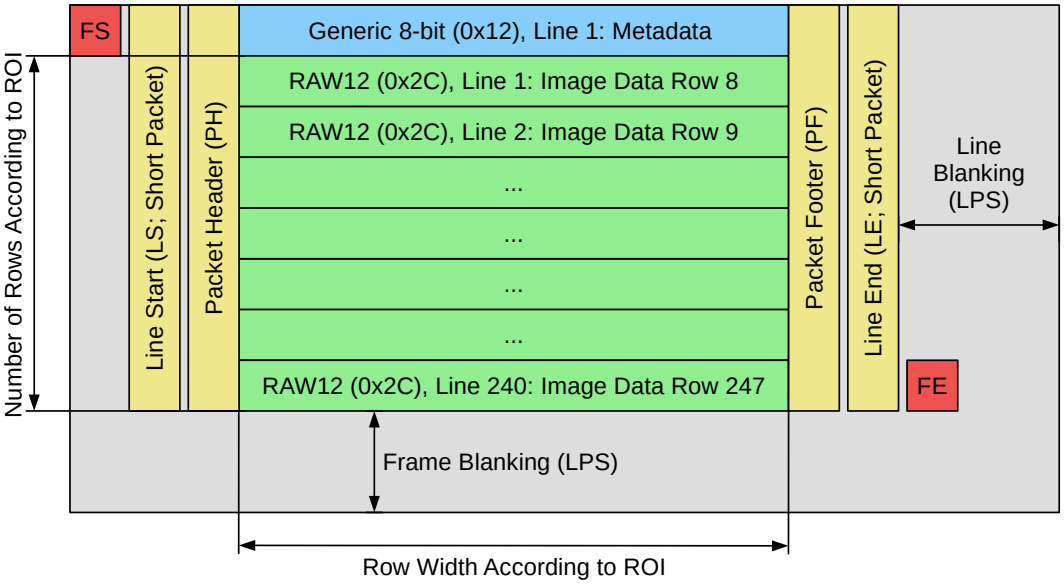

Figure 51: MIPI CSI-2 frame: 1 DCS for default ROI

Figures 52 and 53 illustrate the frame structure for a register dump and a readout sequencer RAM dump, respectively. In both cases, there is only one single MIPI CSI-2 line.

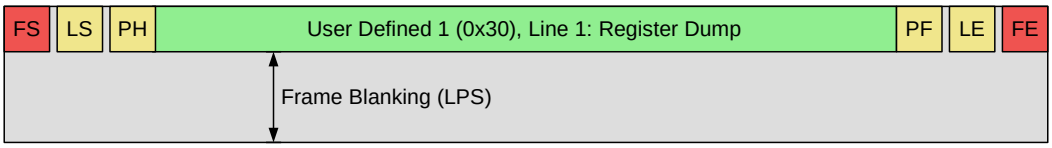

Figure 52: MIPI CSI-2 frame structure for a register dump

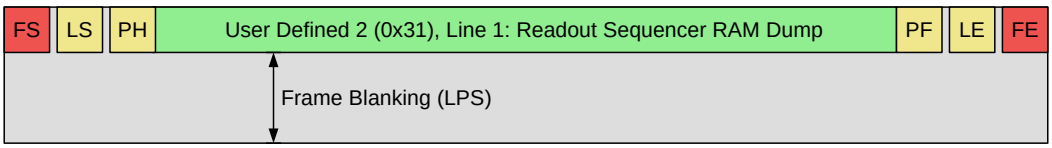

Figure 53: MIPI CSI-2 frame structure for a readout sequencer RAM dump

13.3. Physical layer (MIPI D-PHY)

The packet data described in Chapter 13.2 is all transmitted in high speed (HS) mode. Between the packets, clock and data lanes are going back to low power state (LPS). The physical sequences to switch between HS and LP modes are called start of transmission (SoT) and end of transmission (EoT). Independent of the packet type, a synchronization byte 0xB8 (= 0x1D during transmission with LSB first) is transmitted as part of the SoT sequence.

Figure 54 illustrates the transmission of a MIPI CSI-2 image data frame over D-PHY with SoT and EoT for each packet and LPS between packets. Between frames, all lanes stay in LPS (unless the clock is continuously running).

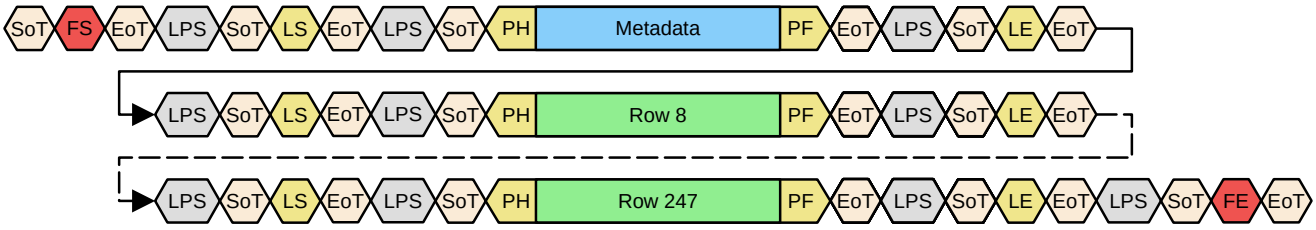

Figure 54: Packet-based transmission of a MIPI CSI-2 image data frame over D-PHY

Figure 55 is depicting the transmission of a long packet over four data lanes (Lx, x ∈ {0, 1, 2, 3}).

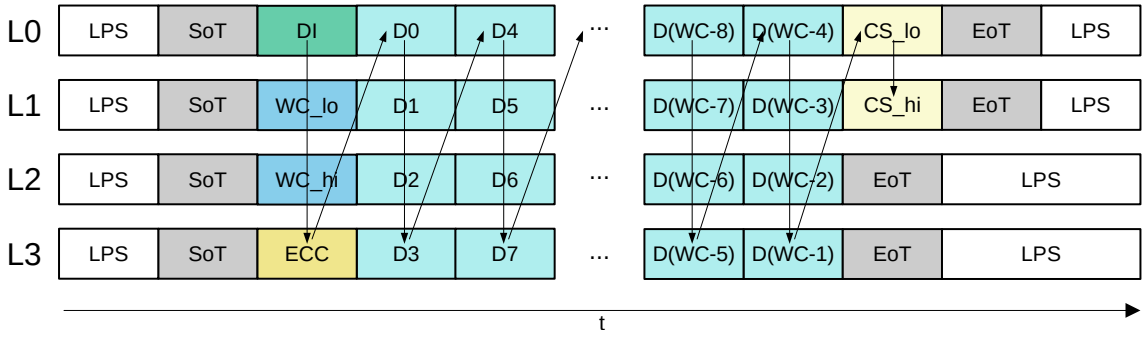

Figure 55: Long packet byte-level data structure in time over four data lanes

The bits per byte are transmitted from least to most significant bit as depicted in Figure 56.

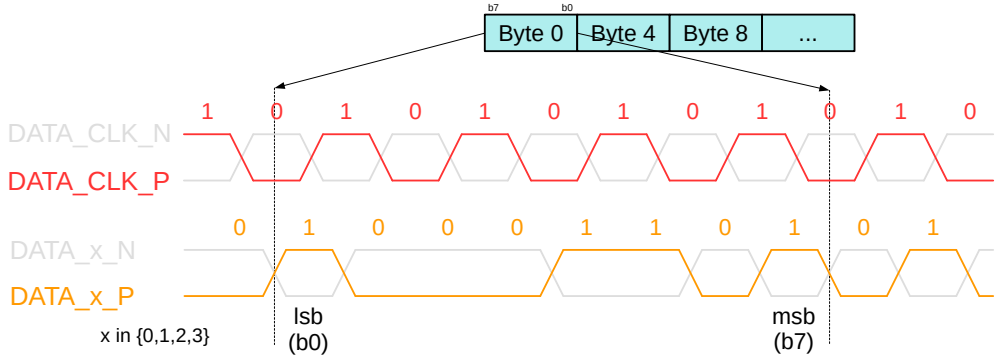

Figure 56: Transmission of a byte over a MIPI D-PHY data lane in high speed mode

14. Hardware implementation

14.1. Typical application diagram

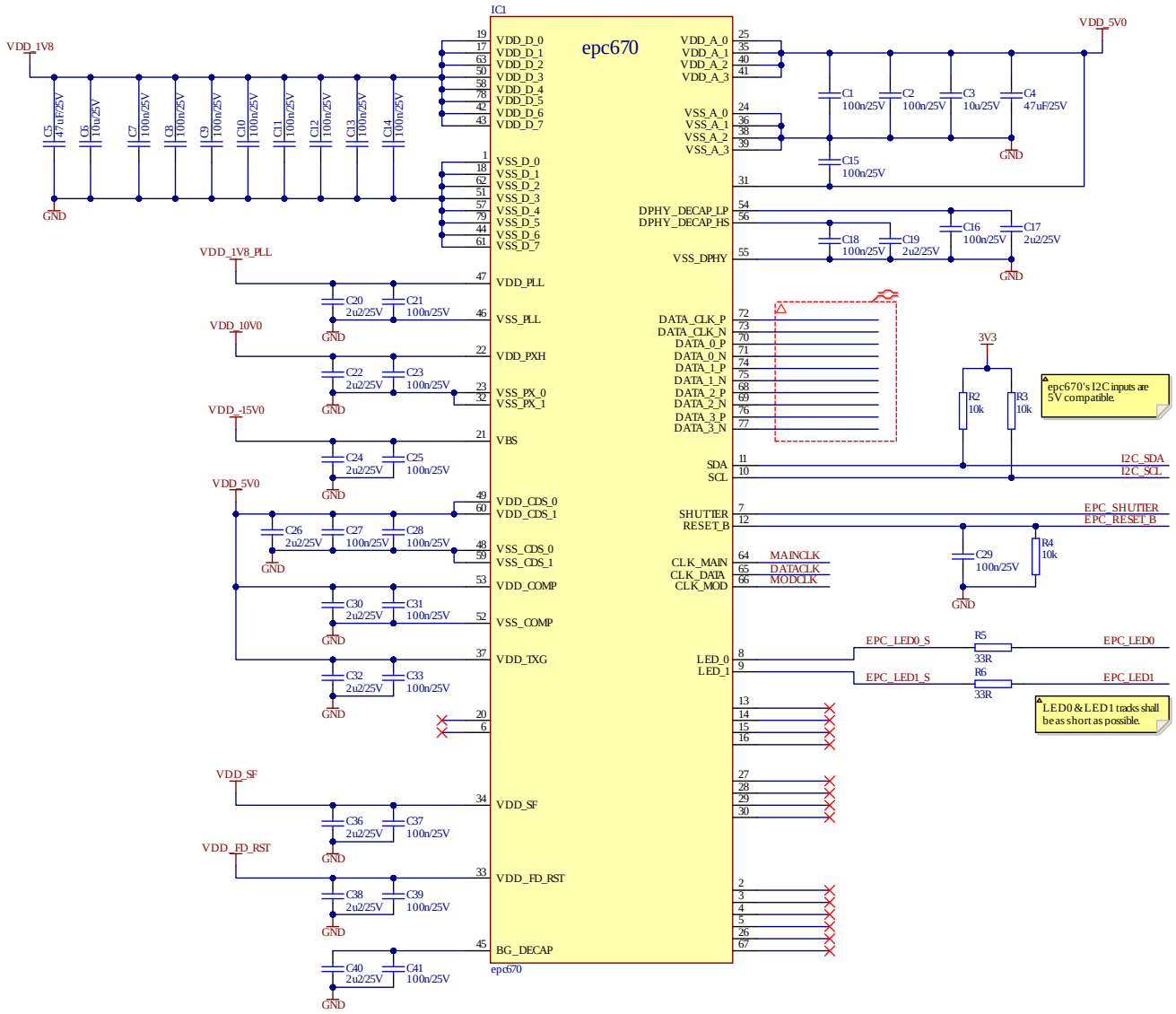

Figure 57: Typical application diagram

14.2. Hardware implementation notes

4. epc670 is supplied with +1.8 V, +5 V, +9 V, +10 V and -15 V. See Figure 57.
5. All VSS shall be connected as directly as possible to a GND plane underneath the imager using vias.
6. Decoupling capacitors must be placed as close as possible to their supply pin pair in order to minimize ripple on the supply rails due to fast switching high-speed signals.
7. +1.8 V is used for supplying the digital logic (VDD_D) and the phase-locked loops (VDD_PLL). Their supply wiring must be separated from the digital wires and physically isolated from each other. Furthermore, good decoupling shall be added – especially for VDD_PLL.
8. The two PLLs are critical parts of the epc670 which directly impact the optical system performance (i.e. distance calculation) and stability. In order to prevent stability issues and unpredictable behavior (e.g. due to timing violations in the digital logic), it is heavily recommended to have a separate LDO for VDD_PLL.
9. +5 V is used for supplying analog blocks of the chip (e.g. band gap or modulation gate drivers).
10. +9 V (VDD_FD_RST) is used for supplying the transistors resetting the floating diffusions (FDs).
11. +10 V is used for supplying the storage gate drivers (VDD_PXH) and the source followers (VDD_SF).
12. -15 V (VBS) is used for biasing the pixel field like reverse-biasing a photodiode. The use of a stable supply source with a low ripple is recommended. There is no switching or active internal circuit-dependent current consumption, except ambient light-dependent leakage current (refer to Table 4, note 3).
13. A 125 MHz external clock source is connected to the CLK_MAIN pin. The frequency accuracy and stability are directly related to the distance readings.

14. A 125 MHz external clock source is connected to the CLK_DATA pin. The frequency accuracy and stability are directly related to the data transmission quality (bit error rate).
15. CLK_MOD input is used as a separate clock source for modulation (LED driver) and demodulation (modulation gate drivers).
16. SCL and SDA are I^2C slave pins which need external pull-up resistors on the PCB. Values of R2 and R3 must be calculated according to the total capacitive load of all I^2C slave/master devices and the operating mode FM or FM+ of the I^2C (Chapter 12) in the application.
17. Track length variation for MIPI clock and data lanes shall be <1 mm.
18. The termination of the MIPI lanes has to be dynamically handled by the connected MIPI D-PHY receiver. In HS mode, 2 mA are flowing on every lane.
19. LED_0/1 pins are push-pull drivers providing symmetric rise/fall times to drive external LED drivers. They are operating in the +1.8 V VDD_D domain. LED_0/1 = LOW (approx. 0 V) corresponds to LED = OFF.

14.3. Application support circuits

The example circuit presented in this chapter is based on a 24 V main supply which is often a good choice to power both the image sensor and the illumination unit. In order to generate all the supplies required by the epc670, a few intermediate supplies need to be generated as shown in Figures 58 and 59.

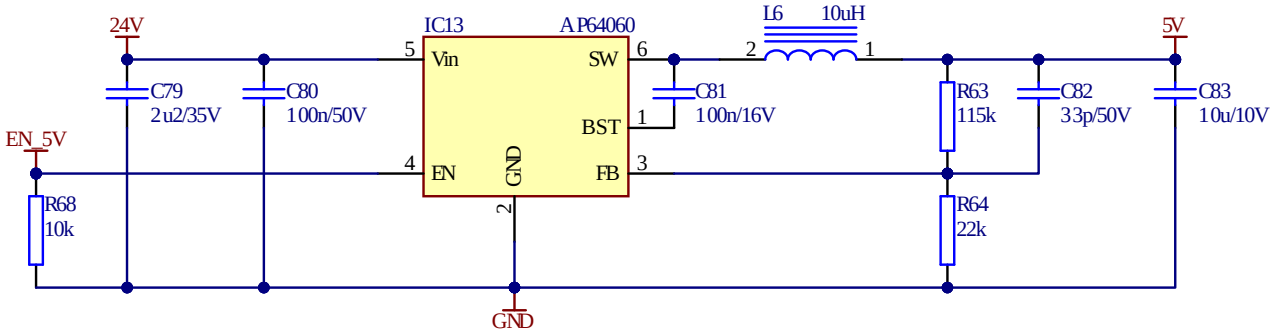

Figure 58: 5 V intermediate supply

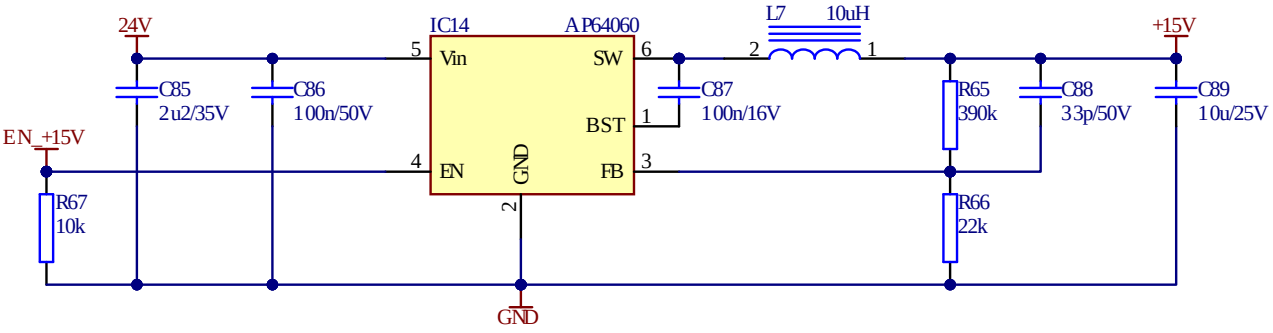

Figure 59: 15 V intermediate supply

14.3.1. Clock source

The epc670 is operated with three individual clocks: main clock, modulation clock, data transmission clock. In order to allow highest flexibility, a clock generator circuit is recommended. Figure 60 shows a possible clock generator circuit which is fully programmable over I2C.

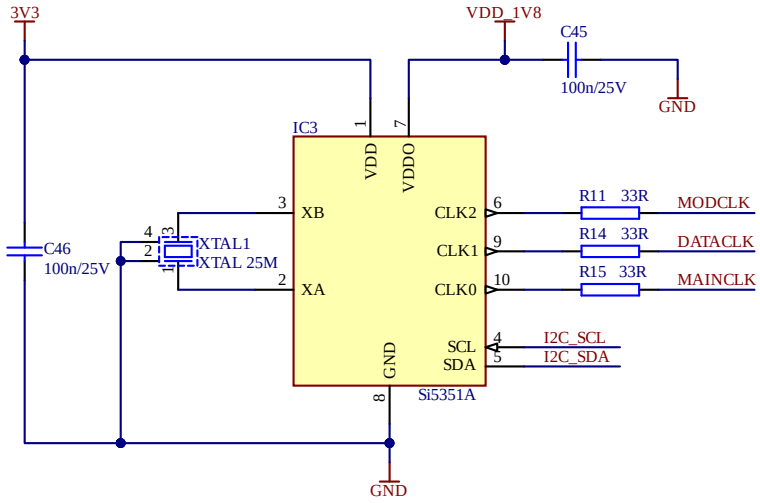

Figure 60: Clock source circuitry proposal

When using the Si5351A clock generator, it is important to make sure that its supplies are enabled in the right order. Otherwise, there might be no clock output. A possible solution for implementing a simple supply sequencing circuit is depicted in Figure 61. Note that an additional 3.3 V supply is required.

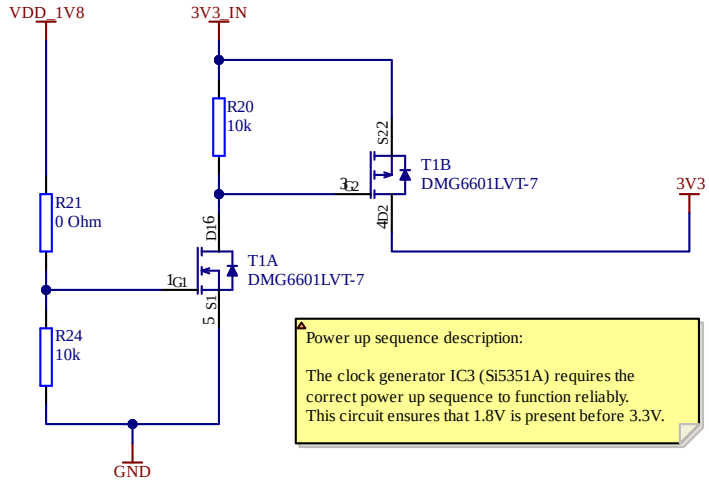

Figure 61: Clock generator supply sequencing

14.3.2. Supply voltage for PLLs

The chip-internal PLLs need a nominal supply voltage of 1.8 V – like the digital core supplied with VDD_1V8. But in order to prevent supply issues for the PLLs, a separate LDO for VDD_PLL shall be used. Figure 62 shows a possible implementation.

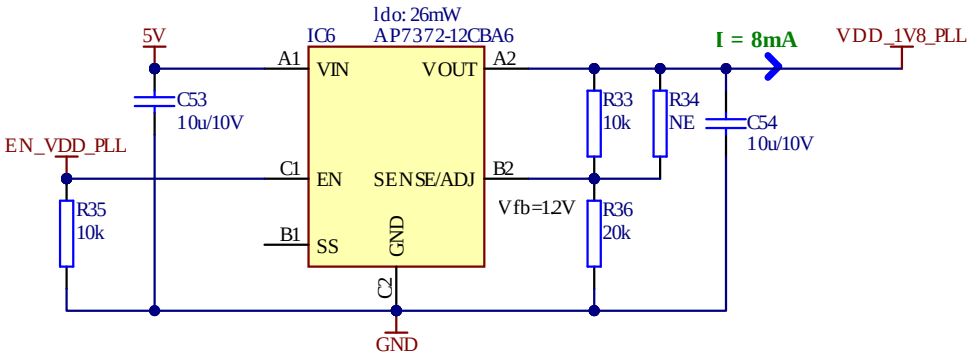

Figure 62: Dedicated LDO for PLL supply

14.3.3. Supply voltage for digital blocks

Since 5 V are already generated by the buck converter introduced in Figure 58, that intermediate supply can be used to derive the 1.8 V supply for the digital core of the epc670. Figure 63 shows a possible implementation.

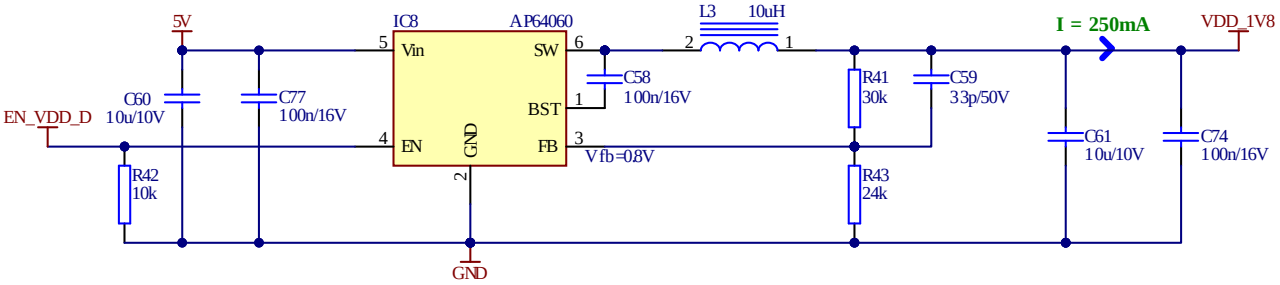

Figure 63: 1.8 V supply circuit example

14.3.4. Supply voltage for analog blocks

Since 5 V are already generated by the buck converter introduced in Figure 58, just an additional inductance and some enable control are recommended to supply VDD_A. Figure 64 shows a possible implementation.

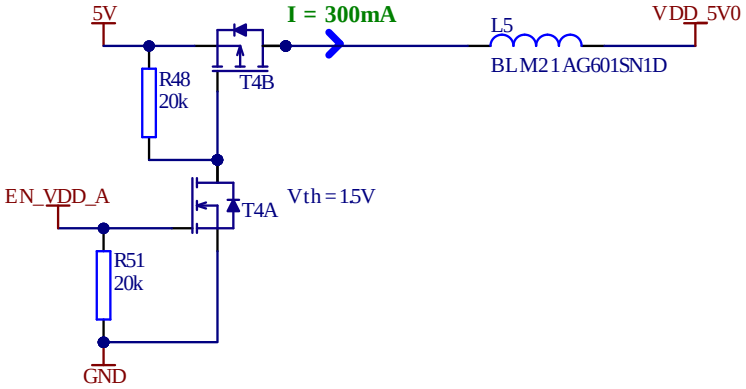

Figure 64: 5.0 V supply circuit example

14.3.5. Supply voltage for source followers

The source followers (SF) need a very low noise supply voltage of nominally 10.0 V. Figure 65 shows a possible implementation. It is to note that the GND net shall be as short as possible connected to the VSS plane of the chip in order to avoid ground noise.

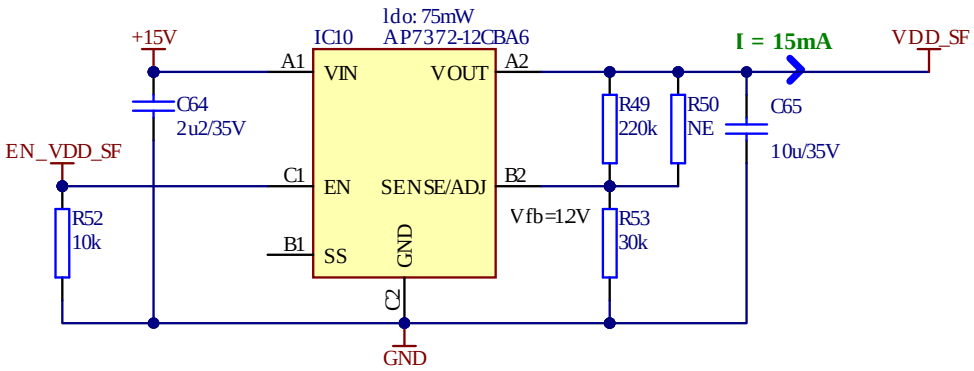

Figure 65: Source follower supply voltage circuit example

14.3.6. Supply voltage for storage gate drivers and pixel field isolation

Since the source followers need a very low noise supply voltage, it is recommended to use a separate supply for VDD_PXH which is powering the storage gate drivers and the pixel field isolation. Figure 66 shows a possible implementation.

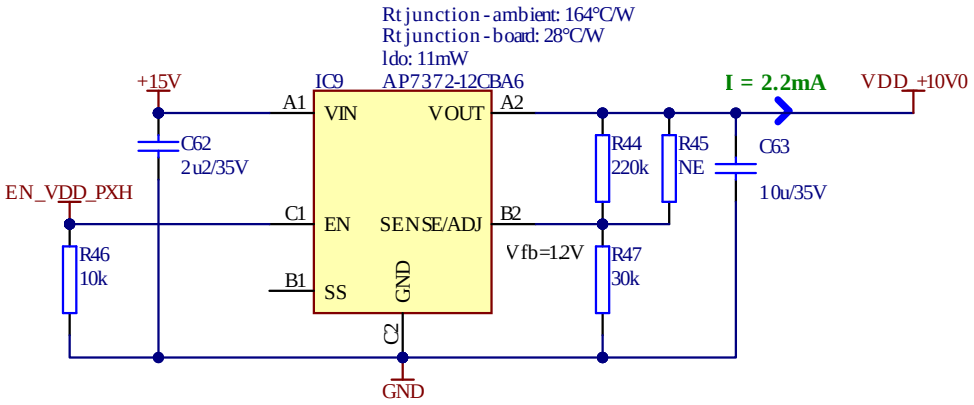

Figure 66: Storage gate driver supply voltage circuit example

14.3.7. Supply voltage for floating diffusion reset transistors

The floating diffusion reset transistors (FD_RST) need a very low noise supply voltage of nominally 9.0 V. Figure 67 shows a possible implementation. It is to note that the GND net shall be as short as possible connected to the VSS plane of the chip in order to avoid ground noise.

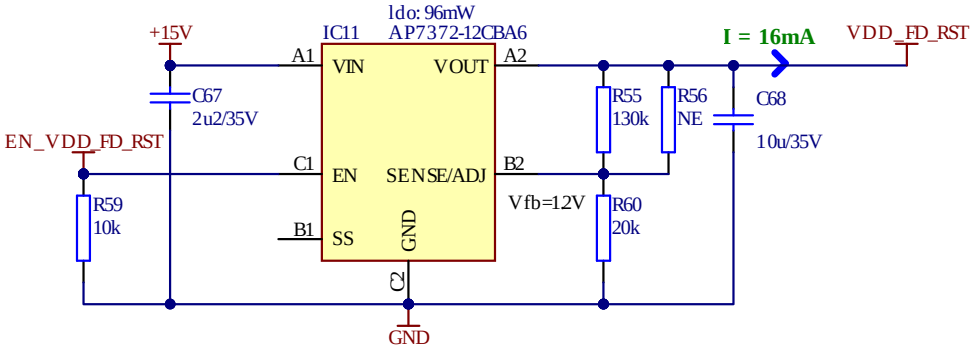

Figure 67: Floating diffusion reset transistor supply voltage circuit example

14.3.8. Supply voltage for imager backside

The imager backside needs to be supplied with a negative voltage VBS in order to guide the excited electrons to the front side where the circuitry is located. Figure 68 shows a possible implementation for the VBS supply.

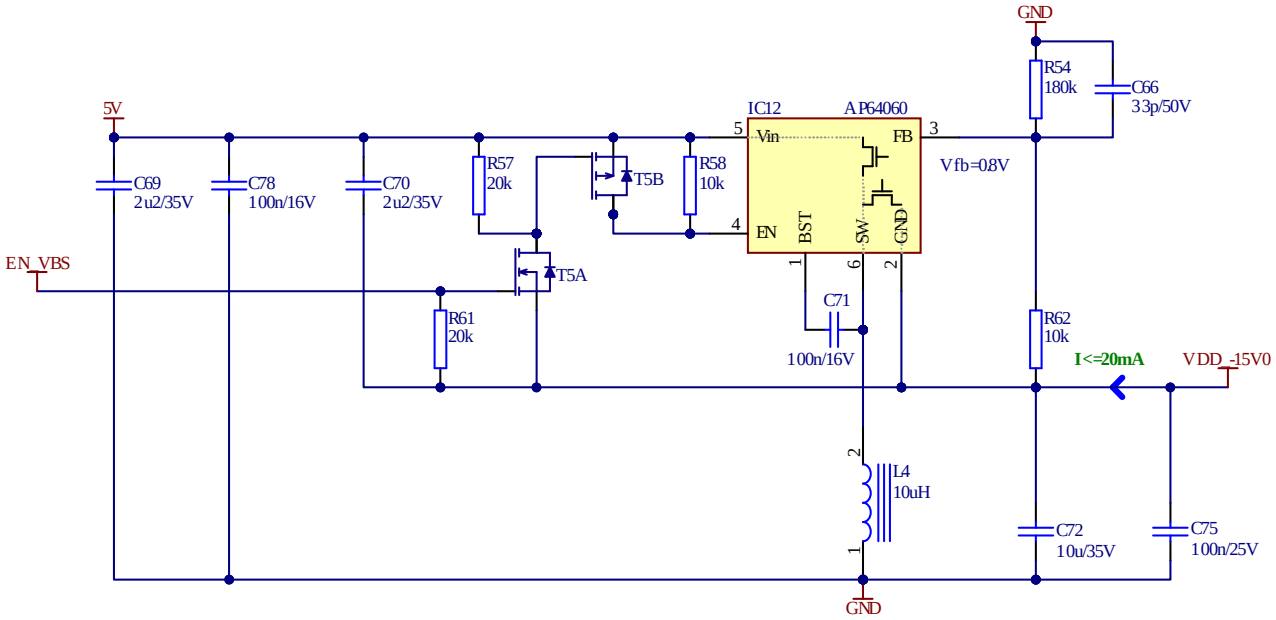

Figure 68: Imager backside supply voltage circuit example

15. Application information

15.1. LED driver

Register LED_CTRL (0x95) can be used to configure illumination control pins LED_0 and LED_1. For example, the polarity can be selected individually in order to match the external LED/LD circuitry used in the application.

IMPORTANT: There are non-modulating DC modes (e.g. grayscale with LED/LD illumination) which keep the LED drivers always turned on. In such cases, the user has to take care that LED drivers and the epc670 chip do not exceed the maximum operating limits.

15.2. Delay line

The modulation control signal for the illumination can intentionally be delayed with the on-chip delay line (DL) in order to add a phase shift between the modulation of the light source and the demodulation of the backscattered light as illustrated in Figure 69. Such a phase shift results in a virtual distance shift.

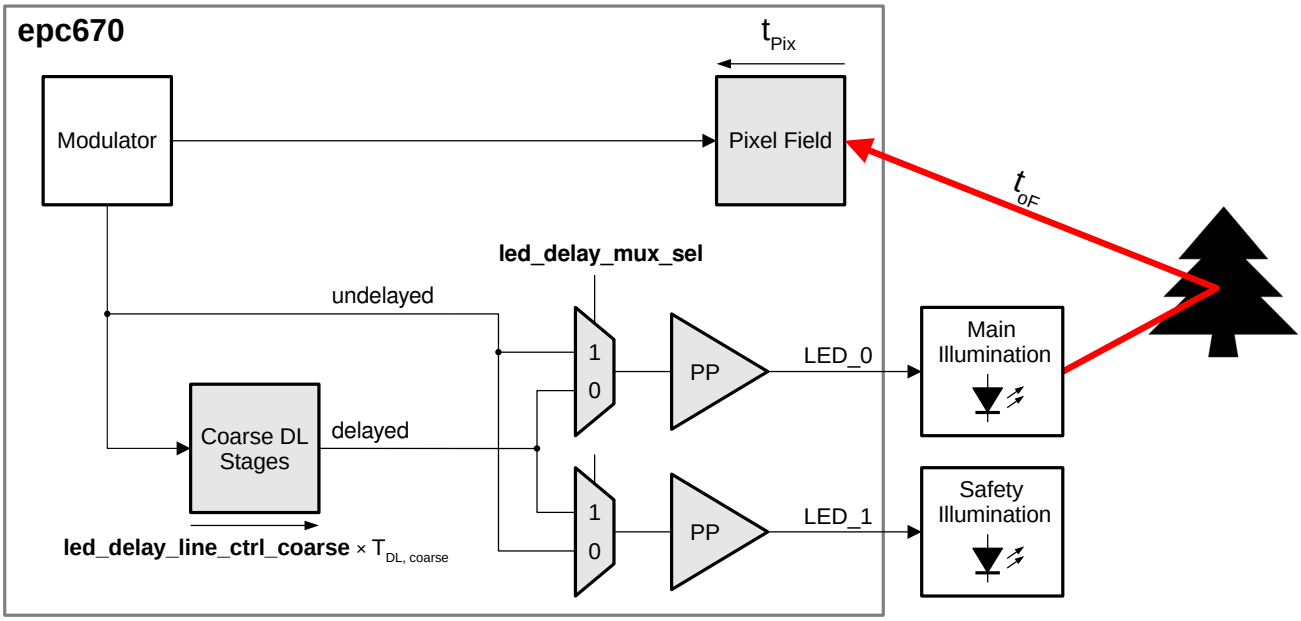

Figure 69: Delay line usage in combination with illumination control pins

Purpose of the delay line

The internal delay line serves two purposes:

1. **Fail-safe test mode of SIL-rated cameras:**
   When connecting a dedicated LED (safety illumination) to the pin LED_1 and placing this LED inside the lens holder, the delay line can be used to check whether the calculated distances shift as expected. This test ensures correct operation of the epc670 ToF imager. Since the LED is directly illuminating the pixel field, no reflection from the scenery is required and all pixels can be tested independent of the scenery. LED_0 shall be temporarily disabled when such a check is done.
2. **Camera calibration:**
   The delay line can be used to virtually step through the distance measurement range of the application. This allows to perform ToF camera calibration in a compact setup.

The delay of the modulation can be adjusted using register LED_DELAY_CTRL_COARSE (0x6F).

Register field **led_delay_mux_sel** of register LED_CTRL (0x95) can be used to select which one of the two LED driver outputs LED_0 and LED_1 shall be delayed. In order to bypass the delay line for one of the LED driver outputs, the delay has to be applied to the other LED driver output.

15.3. 4 DCS burst mode (default)

By default, epc670 is configured to record 4 DCS frames per shutter. Distance and amplitude information is then calculated based on such a DCS burst. This case is illustrated in Figure 70. Often, there will be idle times (gray sections) between DCS bursts due to limited processing capabilities of the microcontroller/CPU or due to eye safety reasons. Nevertheless, it is very important to record a complete DCS burst as fast a possible in order to minimize motion blur artifacts.

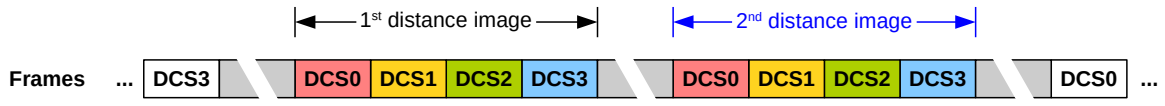

Figure 70: Bursts of 4 DCS frames

15.4. 12 DCS burst HDR mode (recommended)

For various applications, a high dynamic range is required due to distance range requirements or reflectivity variation in the scenery. In such cases, the 12 DCS burst high dynamic range (HDR) mode is a very suitable choice. This operating mode (illustrated in Figure 71) records 4 DCS frames for each of 3 different integration times in a minimum time which allows to minimize motion blur artifacts while maximizing the dynamic range.

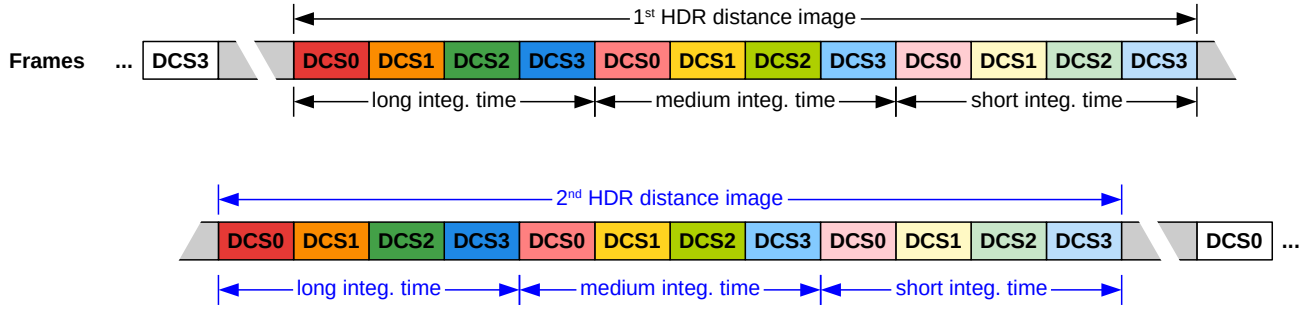

Figure 71: Bursts of 12 DCS frames (4 DCS × 3 integration times)

Implementation example step by step: 12 DCS burst mode using 3 integration times

1. Set the number of frames per shutter to 12: SHUTTER_CTRL = 0xB0 (**dcs_num_sel** = 11)
2. Select integration time from FCT: FCT_CFG_SEL = 0x02 (**cfg_sel_integ_len** = '1')
3. Set long integration time for the first four DCS frames at FCT addresses 0 to 3:
    1. Load FCT entry 0: FCT_ACCESS_CTRL = 0x80 (**fct_addr** = 0x0)
    2. Set long integration time of 2 ms (10 MHz modulation):
        1. FCT_DATA_5 = 0x01
        2. FCT_DATA_4 = 0x38
        3. FCT_DATA_3 = 0x80
    3. Save FCT entry 0: FCT_ACCESS_CTRL = 0xC0 (**fct_addr** = 0x0)
    4. Repeat the previous 3 steps for FTC addresses 1 to 3.
4. Set medium integration time for DCS frames 5 to 8 at FCT addresses 4 to 7:
    1. Load FCT entry 4: FCT_ACCESS_CTRL = 0x84 (**fct_addr** = 0x4)
    2. Set short integration time of 200 µs (10 MHz modulation):
        1. FCT_DATA_5 = 0x00
        2. FCT_DATA_4 = 0x1F
        3. FCT_DATA_3 = 0x40
    3. Save FCT entry 4: FCT_ACCESS_CTRL = 0xC4 (**fct_addr** = 0x4)
    4. Repeat the previous 3 steps for FTC addresses 5 to 7.
5. Set short integration time for DCS frames 9 to 12 at FCT addresses 8 to 11:
    1. Load FCT entry 8: FCT_ACCESS_CTRL = 0x88 (**fct_addr** = 0x8)
    2. Set short integration time of 20 µs (10 MHz modulation):
        1. FCT_DATA_5 = 0x00
        2. FCT_DATA_4 = 0x03
        3. FCT_DATA_3 = 0x20
    3. Save FCT entry 8: FCT_ACCESS_CTRL = 0xC8 (**fct_addr** = 0x8)
    4. Repeat the previous 3 steps for FTC addresses 9 to 11.
6. Issue a shutter (hardware or software) and store the received image data 12 frames).
7. As soon as four DCS frames are available, calculate distance and TOF amplitude for the corresponding integration time (including compensation). When all three distance images are available, merge the distance images by choosing the most reliable distance information based on ToF amplitude and saturation information in order to get the final HDR distance image.
8. Wait for the idle time required to achieve the target frame rate.
9. Repeat steps 6 to 8.

Note: The same concept can also be implemented with only 2 integration times if there is no need for 3 integration times.

15.5. Rolling DCS frames (max. frame rate)

In special applications, it is possible to use all the time the same integration time in continuous distance measurement mode without any grayscale images for ambient-light compensation. Such a set-up allows enhancing the distance measurement rate by a factor of 4 by using rolling DCS frames.

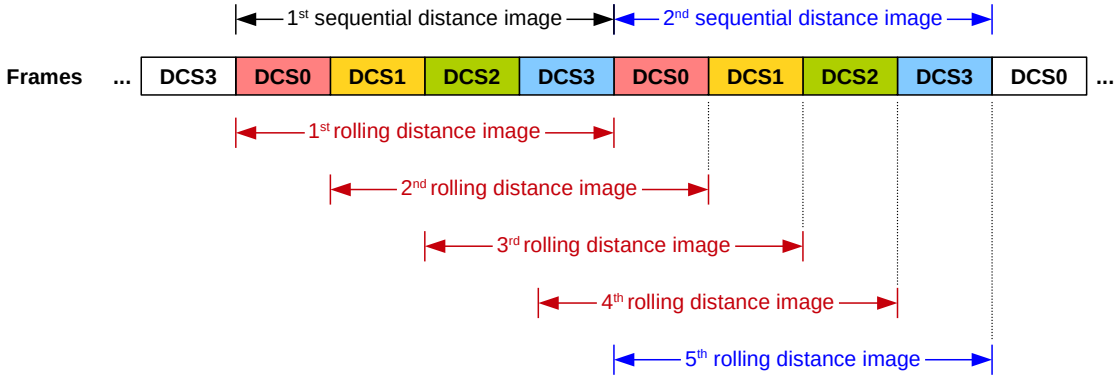

Figure 72: Rolling DCS frames

As shown in Figure 72, the algorithm performs with each new DCS frame a new distance calculation based on the new and last three DCS frames.

15.6. Rolling DCS frames in HDR mode

epc670 allows to individually configure every single DCS. This opens also the possibility to acquire multiple DCS frames of the same phase with different integration times before recording the next phase for example. The high dynamic range (HDR) rolling mode makes use of this. It combines high dynamic range with maximum frame rate.

The resulting HDR distance frames will be available in an equidistant manner over time, e.g. one HDR distance image every 6.7 ms (150 fps).

The example illustrated in Figure 73 is using two integration times:
50 µs for detecting short range objects and 2 ms doing the same for the long range.

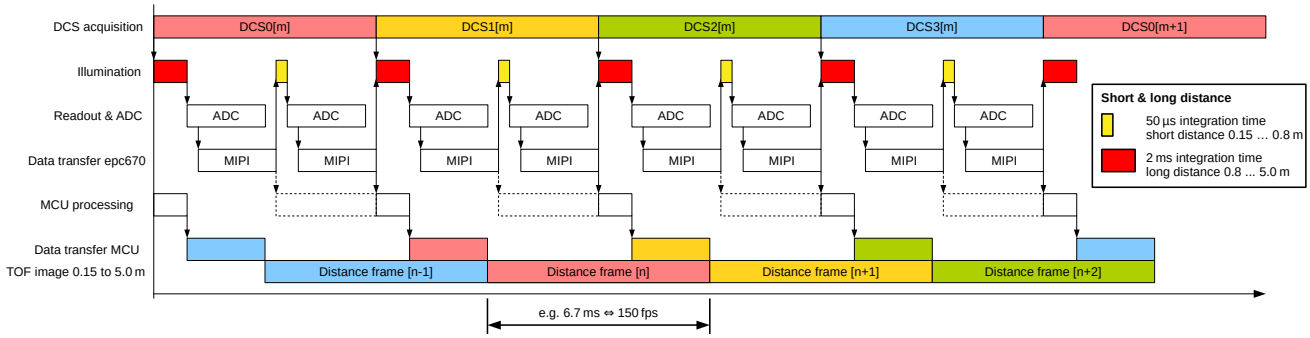

Figure 73: HDR rolling mode sequence (two integration times)

Implementation example step by step: rolling mode using 2 integration times

1. Set the number of frames per shutter to 2: SHUTTER_CTRL = 0x10 (**dcs_num_sel** = 1)
2. Select integration time from FCT: FCT_CFG_SEL = 0x02 (**cfg_sel_integ_len** = '1')
3. Set long integration time and the first phase to be recorded for the first DCS at FCT address 0:
    1. Load FCT entry 0: FCT_ACCESS_CTRL = 0x80
    2. Set long integration time of 2 ms (10 MHz modulation):
        1. FCT_DATA_5 = 0x01
        2. FCT_DATA_4 = 0x38
        3. FCT_DATA_3 = 0x80
    3. Select phase DCS0 for first frame: FCT_DATA_15 = 0x00
    4. Save FCT entry 0: FCT_ACCESS_CTRL = 0xC0
4. Set short integration time and the first phase to be recorded for the second DCS at FCT address 1:
    1. Load FCT entry 1: FCT_ACCESS_CTRL = 0x81
    2. Set short integration time of 50 µs (10 MHz modulation):
        1. FCT_DATA_5 = 0x00
        2. FCT_DATA_4 = 0x07
        3. FCT_DATA_3 = 0xD0

3. Select phase DCS0 for second frame: FCT_DATA_15 = 0x00
4. Save FCT entry 1: FCT_ACCESS_CTRL = 0xC1

5. Issue a shutter (hardware or software) and store the received image data.
6. As soon as four DCSs are available, calculate distance and TOF amplitude for both integration times based on the latest four DCS frames (including compensation). Merge the two distance images by choosing the most reliable distance information based on ToF amplitude and saturation information in order to get the final HDR distance image.
7. Select the phase to be recorded next for FCT addresses 0 and 1:
    1. Load FCT entry 0: FCT_ACCESS_CTRL = 0x80
    2. Select phase DCSZ for first frame: FCT_DATA_15 = 0xZ0
    3. Save FCT entry 0: FCT_ACCESS_CTRL = 0xC0
    4. Load FCT entry 1: FCT_ACCESS_CTRL = 0x81
    5. Select phase DCSZ for second frame: FCT_DATA_15 = 0xZ0
    6. Save FCT entry 1: FCT_ACCESS_CTRL = 0xC1
8. Wait for the idle time required to achieve the target frame rate.
9. Repeat steps 5 to 8.

Note: The same concept can also be implemented with 3 integration times if more dynamic range is required.

16. Terms, definitions and abbreviations

Table 31: Definitions and abbreviations

Abbreviation	Term, Definition	Explanation
ABS	Ambient Light Suppression	
ADC	Analog Digital Converter	
CDS	Correlated Double Sampling	Noise reduction technique where a known reset value is subtracted from the measured value to get the actual signal value.
CGU	Clock Generation Unit	
CRC	Cyclic Redundancy Check	CRC codes are basically remainders of polynomial divisions. They can be used for error detection and correction in digital data.
CSP	Chip Scale Package	
DCS	Differential Correlation Sample	
DL	Delay Line	
fps	Frames per Second	
HDR	High Dynamic Range	
HW	Hardware	
IC	Integrated Circuit	
LED/LD	Light Emitting Diode / Laser Diode	
LSB	Least Significant Bit	
MGa	Modulation Gate A	
MGb	Modulation Gate B	
MGx	Modulation Gate A or B	
mga	MGa control signal	
mgb	MGb control signal	
mgx	MGx control signal	
MSB	Most Significant Bit	
PLL	Phase-Locked Loop	
ROI	Region of Interest	Defines the part of the pixel field which is used for imaging.
QVGA	Quarter VGA	320 × 240 pixels resolution
SW	Software	
SGa	Storage Gate A	
SGb	Storage Gate B	
SGx	Storage Gate A or B	
TOF	Time of Flight	

17. IMPORTANT NOTICE

ESPROS Photonics AG and its subsidiaries (ESPROS) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to ESPROS' terms and conditions of sale supplied at the time of order acknowledgment.

ESPROS warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with ESPROS' standard warranty. Testing and other quality control techniques are used to the extent ESPROS deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

ESPROS assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using ESPROS components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

ESPROS does not warrant or represent that any license, either express or implied, is granted under any ESPROS patent right, copyright, mask work right, or other ESPROS intellectual property right relating to any combination, machine, or process in which ESPROS products or services are used. Information published by ESPROS regarding third-party products or services does not constitute a license from ESPROS to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from ESPROS under the patents or other intellectual property of ESPROS.

Resale of ESPROS products or services with statements different from or beyond the parameters stated by ESPROS for that product or service voids all express and any implied warranties for the associated ESPROS product or service. ESPROS is not responsible or liable for any such statements.

ESPROS products are not authorized for use in safety-critical applications (such as life support) where a failure of the ESPROS product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of ESPROS products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by ESPROS. Further, Buyers must fully indemnify ESPROS and its representatives against any damages arising out of the use of ESPROS products in such safety-critical applications.

ESPROS products are neither designed nor intended for use in military/aerospace applications or environments unless the ESPROS products are specifically designated by ESPROS as military-grade. Only products designated by ESPROS as military-grade meet military specifications. Buyers acknowledge and agree that any such use of ESPROS products which ESPROS has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

ESPROS products are neither designed nor intended for use in automotive applications or environments unless the specific ESPROS products are designated by ESPROS as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, ESPROS will not be responsible for any failure to meet such requirements.