User’s guide to BGT60LTR11AiP

60 GHz radar

About this document

Scope and purpose

This application note explains in detail how to use BGT60LTR11AiP in an end application.

The reader will find:

- Explanation of all of the different building blocks
- How to operate the different blocks
- Settings of the serial peripheral interface (SPI) registers grouped by topic, including truth tables

Intended audience

Hardware and software engineers working on designs with Infineon’s BGT60LTR11AiP.

Table of contents

About this document **1**
Table of contents **2**
1 Main controller **4**
1.1 Quad-state inputs 4
1.1.1 QS1 5
1.1.2 QS2 5
1.1.3 QS3 5
1.1.4 QS4 5
1.2 Power-up and sequencing 6
1.2.1 Power-up 6
1.2.2 Init sequence 7
1.2.3 Pulsed sequence 9
1.2.4 CW sequence 12
1.2.5 SPI mode sequence 12
1.2.6 Detector 13
1.3 Overview of dynamic control signals 13
2 SPI **15**
2.1 SPI description 15
2.1.1 SPI write mode 15
2.1.2 SPI read mode 15
2.1.3 SPI burst mode 16
2.1.3.1 SPI burst access 16
2.2 SPI register 17
2.2.1 Register overview 17
2.2.2 Direct access register 18
2.2.3 Register map bitfields 18
2.2.4 Register reg0 – direct access register 20
2.2.5 Register reg1 – direct access register 21
2.2.6 Register reg4 – PLL config 1 23
2.2.7 Register reg5 – PLL config 2 25
2.2.8 Register reg6 – PLL config 3 26
2.2.9 Register reg7 – duty cycling, timing, PD, MPA 27
2.2.10 Register reg8 – divider 29
2.2.11 Register reg9 – BB 30
2.2.12 Register reg10 – quad-state inputs 31
2.2.13 Register reg11 – BB detector 32
2.2.14 Register reg12 – BITE 33
2.2.15 Register reg15 – digital control 35
2.2.16 Register reg34 – ADC start 35
2.2.17 Register reg35 – ADC convert 36
2.2.18 Register reg36 – ADC status 37
2.2.19 Register reg38–53 – ADC result 38
2.2.20 Register reg55 – e-fuses 38
2.2.21 Register reg56 – status and chip version 40
2.2.22 Register GSR0 – SPI status register 41
3 Analog to digital converter **42**
3.1 ADC conversion sequence 42

Table of contents

3.1.1 Enable bandgap 42
3.1.2 Enable local ADC clock 42
3.1.3 Enable ADC 42
3.1.4 Start ADC conversion 42
3.1.4.1 Single conversion 42
3.1.4.2 Sequential conversion 42
3.2 ADC configuration 43
3.2.1 Analog input channel gain 43
3.2.2 Analog input voltage sampling 43
3.2.3 ADC phases 43
3.3 Conversion time 43
3.4 ADC power-down sequence 43
4 Detector 45
4.1 Detector block diagram 45
4.2 Schmitt trigger 45
4.3 Pulse generator 45
4.4 Digital evaluation 45
4.4.1 Holdtime 46
4.4.2 Detector hit counter setting 46
4.4.3 Detector hit counter reset setting 46
4.4.4 Detector direction counter hit setting 46
4.4.5 Example wave diagram with outputs and counters 47
Revision history 47

1 Main controller

The purpose of the main controller is to handle pulsed and continuous mode autonomously. Additionally, there is SPI mode available, where everything is controllable from an external microcontroller using the SPI.

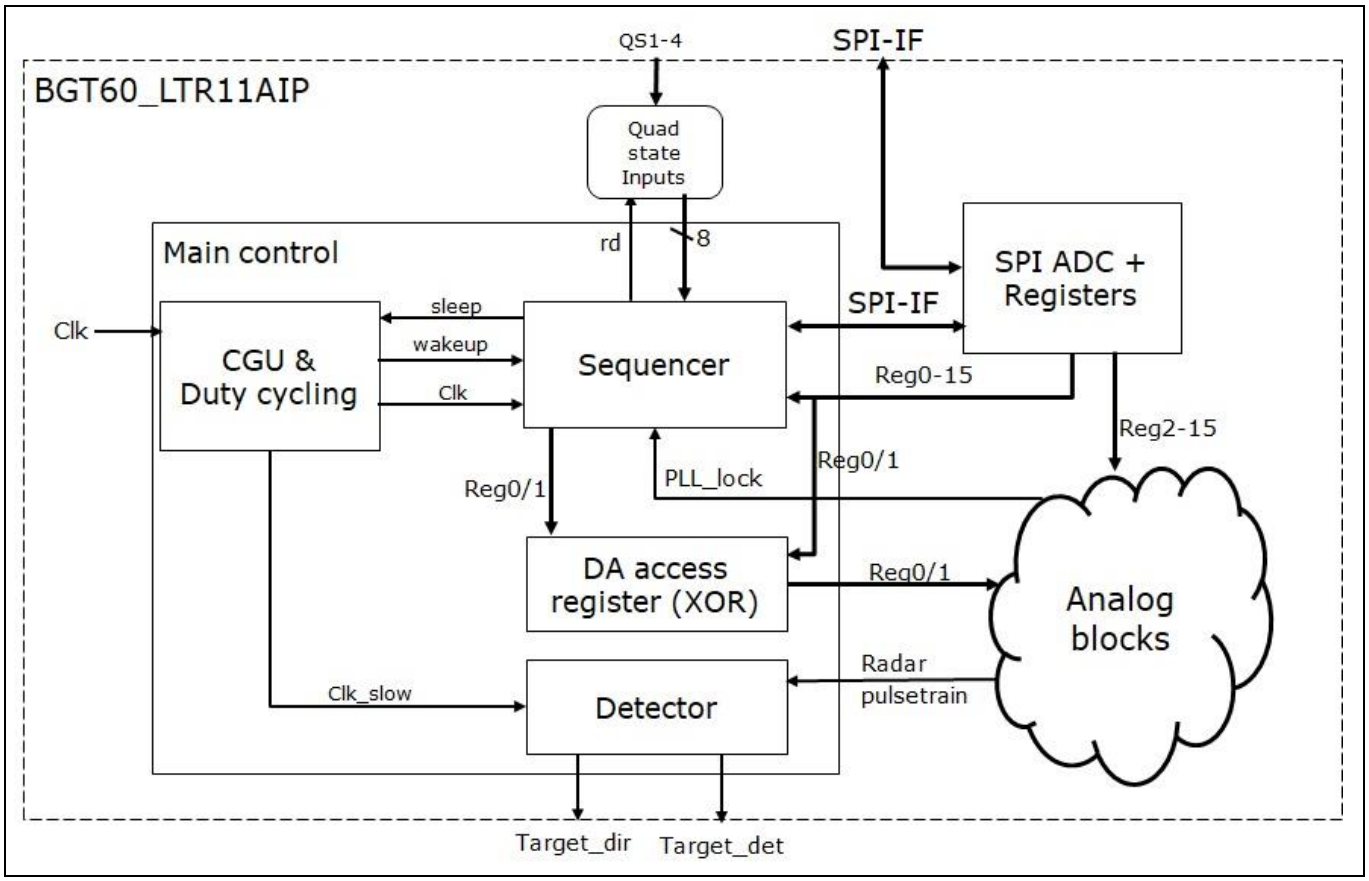

Figure 1 Main controller block diagram

BGT60LTR11AiP provides four quad-state inputs, QS1–4. With one quad-state input it is possible to get four states from one input pin. These pins are used for configuration of the chip.

Furthermore, there are 16 e-fuses available, which are also used for configuration.

1.1 Quad-state inputs

The quad-state inputs enable configuration of four different states with one input pin. Table 1 shows possible input states and the resulting internal signals in binary description. Quad-state inputs are sampled at the start of the init sequence by the internal main controller at power-up. A change after this sampling has no effect. Resampling can be triggered by setting the reset pin or activating the soft reset by writing the corresponding bit in register 15.

Table 1 **States of a quad-state input**

Pad	b1	b0
Ground	0	0
Open	0	1

100 kΩ to V_{DD}	1	0
V_{DD}	1	1

1.1.1 QS1

QS1 is used to select the mode of the chip. Countinuous wave mode makes the chip radiate all the time. Autonomous mode is controlled by the chip's internal state machine – with a default on-time of 5 µs and pulse reparation time (PRT) of 500 µs, the power consumption can be reduced. SPI mode offers full control of the chip. When synchronization is needed between radar and microcontroller, SPI mode with external 9.6 MHz clock-enabled mode can be selected, and the 9.6 MHz clock output can be used for the microcontroller's clock source.

Table 2 **QS1**

Pad	b1	b0	Operating mode
Ground	0	0	Continuous wave (CW) mode
Open	0	1	Autonomous pulsed mode
100 kΩ to V_{DD}	1	0	SPI mode with external 9.6 MHz clock enabled
V_{DD}	1	1	SPI mode

1.1.2 QS2

QS2 is used to select the detector comparator threshold voltage – it is written into the bitfield bb_det_thrs of register 10 described in Table 32.

Table 3 **QS2**

Pad	b1	b0	Detector comparator threshold
Ground	0	0	787.5 mV
Open	0	1	937.5 mV
100 kΩ to V_{DD}	1	0	1087.5 mV
V_{DD}	1	1	1237.5 mV

1.1.3 QS3

QS3 is used to select the holdtime of the T_{det_o} output – it is written into the bitfield bb_det_hold of register 10 described in Table 32.

Table 4 **QS3**

Pad	b1	b0	target_det holdtime
Ground	0	0	10 ms
Open	0	1	1 s
100 kΩ to V_{DD}	1	0	10 s
V_{DD}	1	1	1 min.

1.1.4 QS4

QS4 is used to select the device operating frequency by configuring the PLL. Frequency is also dependent on the Japan e-fuse.

Table 5 **QS4**

Pad	b1	b0	Japan e-fuse	VCO frequency
Ground	0	0	1	61.1 GHz
Open	0	1	1	61.2 GHz
100 kΩ to V_{DD}	1	0	1	61.3 GHz
V_{DD}	1	1	1	61.4 GHz
Ground	0	0	0	60.6 GHz
Open	0	1	0	60.7 GHz
100 kΩ to V_{DD}	1	0	0	60.8 GHz
V_{DD}	1	1	0	60.9 GHz

1.2 Power-up and sequencing

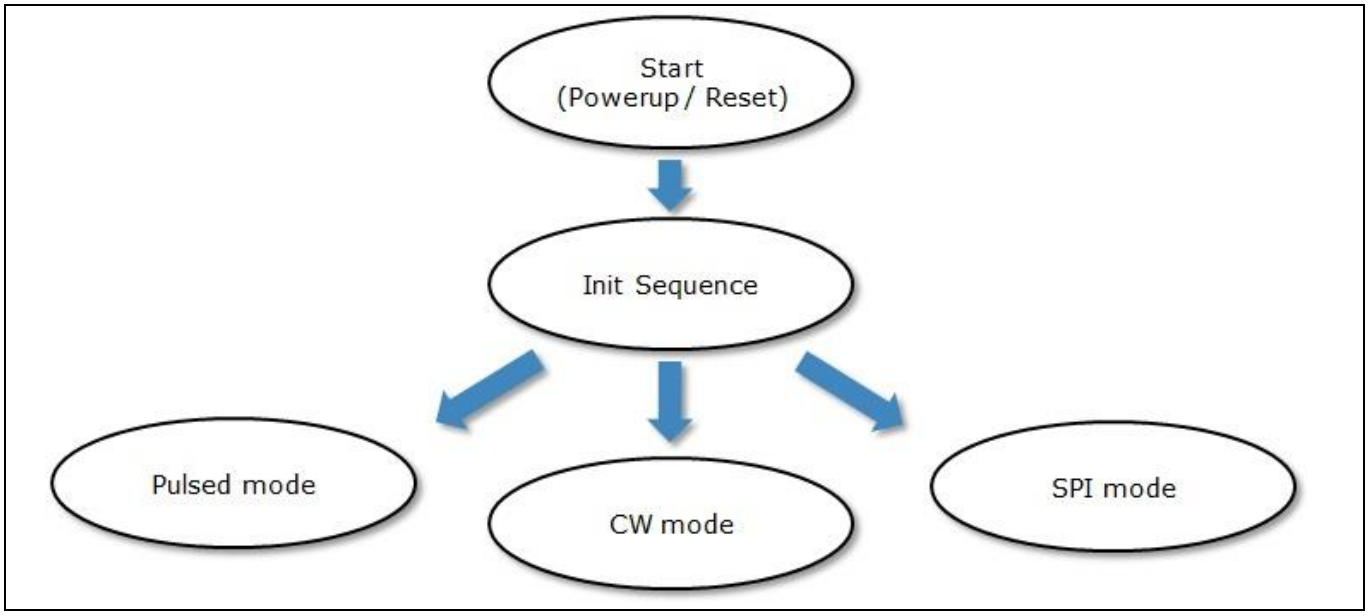

Figure 2 State diagram overview

Depending on the QS1 settings, one of three available modes is selected. At start-up the internal main controller has control over the SPI. So it is not possible to program the chip externally.

During the init sequence, after writing the defaults into the registers, the main controller hands over control to the optional external controller. The main controller doesn't use the SPI after the init sequence. SPI accesses can happen at any time, but they are ignored as long as the main controller still has control over the SPI.

Furthermore, the main controller is halted as long as the pad spi_cs_n_i is active (=0), to prevent synchronization problems. This is independent of the current master of the SPI. So the pad must be set to 1 if SPI is not used or an external controller does not exist.

1.2.1 Power-up

Figure 3 shows internal signals relevant for power-up.

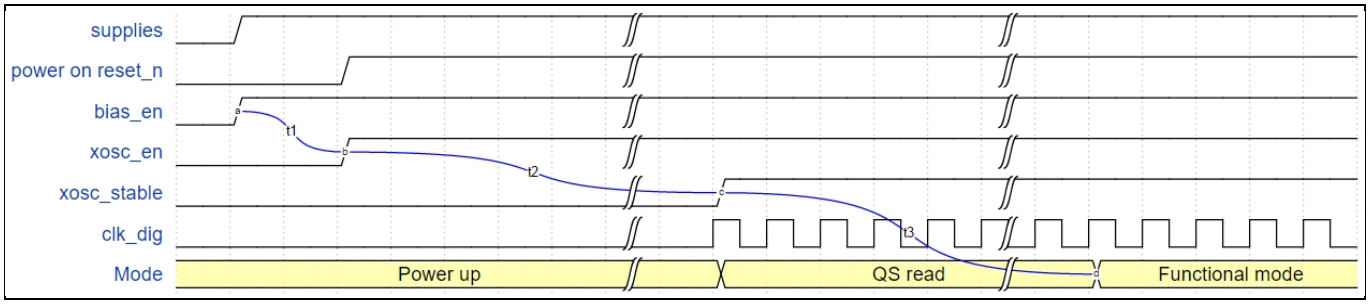

Figure 3 Power-up

Supply ramp needs to be shorter than 400 µs. Bias_en is connected directly to the supply, and therefore it ramps simultaneously. The integrated power-on reset ensures the digital parts wake up in a defined state and this signal is also connected to xosc_en, which starts up the oscillator. Time t_1 between rising edges of these signals should be at least 9 µs. This is fulfilled when the time for ramping supply is as defined.

The oscillator needs the time t_2 to become stable and activate the clock for the main controller; t_2 is smaller than 1 ms. The time t_3 is needed for reading the configuration inputs QS1–4. This takes 200 µs. The chip is now able to accept external SPI commands in SPI mode; in pulsed mode or CW mode it takes 25 µs more.

1.2.2 Init sequence

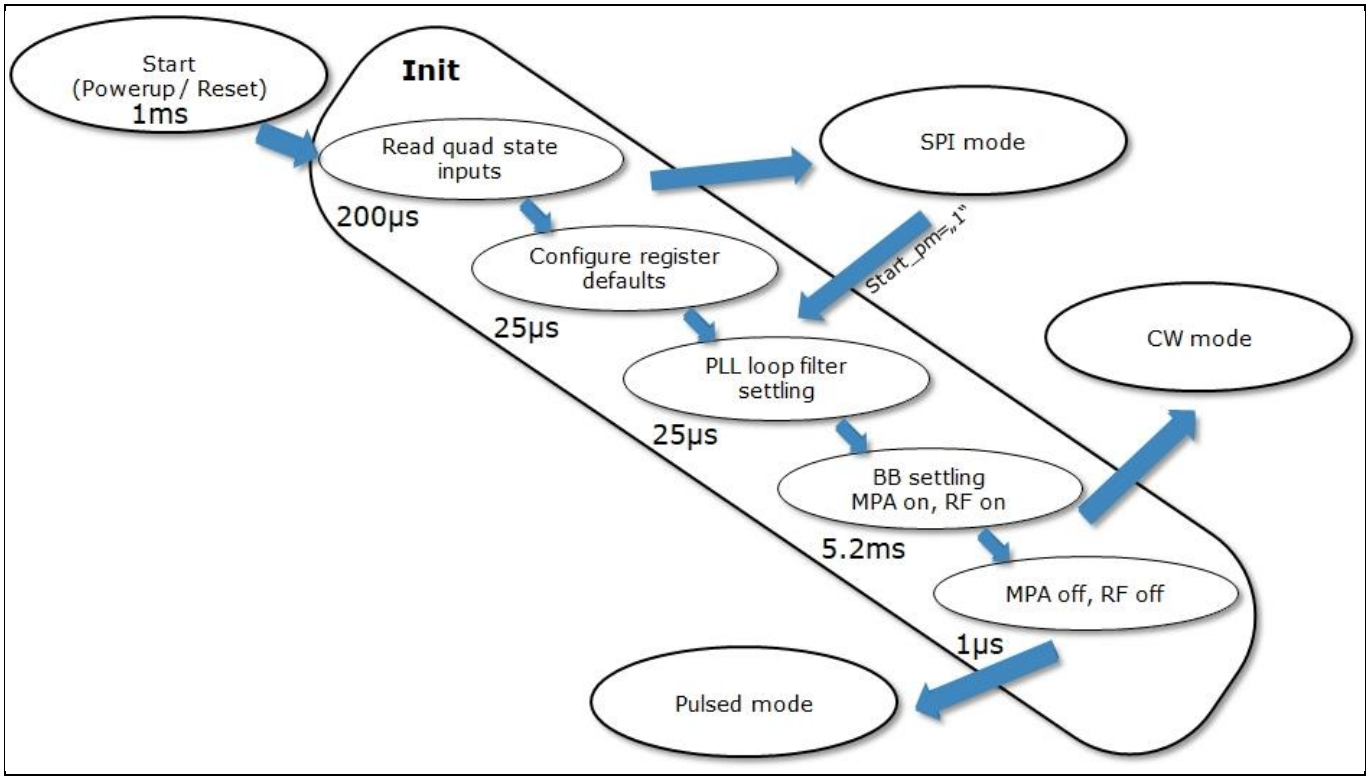

Figure 4 Init sequence

The init sequence starts directly after power-up/reset. It consists of the following steps:

1. Read quad-state inputs.
   Quad-state inputs need 200 µs for analog settling; during waiting also the e-fuses are read.
   After reading the selected mode is known.

If it is SPI mode, init sequence is terminated and control over the SPI is handed over to the external microcontroller. The main controller switches to SPI mode.

2. For pulsed and CW mode now the default values for the configuration registers are prepared and written using the SPI of the SPI ADC block. Then control over the SPI is handed over to the external microcontroller but init sequence continues.
3. PLL is started; medium power amplifier (MPA) is not activated during the loop filter settlement. After 20 µs the MPA is also activated, and RF is also running. The next 5 ms are used for baseband (BB) settling.
4. If CW mode is selected, init sequence is terminated and the main controller switches into CW mode.
5. For pulsed mode MPA, RF and PLL are switched off and the main controller switches to pulsed mode.

The status bit init_done (reg56[14]) is set when leaving init sequence.

QS4 is used to select the device operating frequency by configuring the PLL. Frequency is also dependent on the Japan e-fuse.

Table 6 **Init sequence in detail**

No.	Command	Description
1	Write reg1 0x0100	Set bit qs_rd_en
2	2*read reg55	Read fuse values, two times to provide enough clocks to read out fuses within SPI ADC before reading the register
3	Wait 192 µs	Wait 200 µs before reading quad-state inputs
4	Read quad-state inputs	Read and end init sequence if mode = SPI mode
5	Write reg1 0x0000	Reset bit qs_rd_en
6	SPI write reg4 reg4_init	Write defined default value (partly calculated from e-fuses)
7	SPI write reg5 reg5_init	Write defined default value (partly calculated from e-fuses)
8	SPI write reg6 reg6_init	Write defined default value
9	SPI write reg7 reg7_init	Write defined default value (partly calculated from e-fuses)
10	SPI write reg8 reg8_init	Write defined default value (partly calculated from e-fuses)
11	SPI write reg9 reg9_init	Write defined default value (partly calculated from e-fuses)
12	SPI write reg10 reg10_init	Write defined default value (calculated from quad-states)
13	SPI write reg11 reg11_init	Write defined default value (partly calculated from e-fuses)
14	Write reg0 0x311F	Set vco_buf_en, vco_en, pll_en, rx/txbuf_en, mixIQ_en, lna_en
15	Write reg1 0x1036	Set div_bias_en, bb_boost_dis, bb_clk_chop_en, bb_strup_hp, bb_amp_en
16	Wait PLLen 2 PLLactive	Wait defined time pll_en to pll_active (2 µs)
17	Write reg0 0x371F	Set pll_active, pll_clk_gate_en
18	Wait for lock detect	Wait and set control over SPI to external
19	Wait 20 µs	PLL loop filter settling with MPA off

No.	Command	Description
20	Write reg0 0x373F	Set mpa_en
21	Wait MPA 2 sample enable	Wait defined time MPA enable to sample and hold – mpa2sh_dly (reg7[5:4])
20	Write reg1 0x1037	Set bb_sample_en
21	Wait 5 ms	Wait time for BB settling
22	End 4 CW mode	End init sequence if mode = CW mode
23	Write reg1 0x1092	Reset bb_boost_dis, bb_strup_hp, bb_sample_en, set bb_dig_det_en Wait 100 ns
24	Write reg0 0x371F	Reset mpa_en
25	Wait 20 µs	Allow settling of PLL without active MPA for best relocking in pulsed mode
26	Write reg0 0x311F	Reset pll_active, pll_clk_gate_en Wait 100 ns
26	Write reg1 0x0092	Reset div_bias_en
27	Write reg0 0x0900	RF off, only pll_en is still on and pll_open_loop is set
28	End 4 pulsed mode	End init sequence for pulsed mode, pulsed mode starts with sleep phase defined by dc_rep_rate (reg7[11:10]). Afterward the pulsed sequence is started.

1.2.3 Pulsed sequence

In pulsed mode the device is active only a short time followed by a time where VCO, RF and PLL are off. BB and detector keep running all the time. On/off rate can be configured and is in the range from about 1:5 up to 1:140. Default setting is 1:35. This is due to a default repetition time of 500 µs and an on-time of 14 µs, which consists of about 9 µs needed for start-up of RF and PLL + 5 µs default sample time.

Table 7 **Pulsed sequence in detail**

No.	Command	Description
1	Write reg0 0x391F	Set vco_buf_en, vco_en, pll_en, rx/txbuf_en, mixIQ_en, lna_en
2	Write reg1 0x1092	Set div_bias_en
3	Wait VCO 2 PLL	Wait defined time VCO on to PLL on – vco2pll_dly (reg7[6])
4	Write reg0 0x3F1F	Set pll_active, pll_clk_gate_en
5	Wait for lock detect	
6	Write reg0 0x3F3F	Set mpa_en
7	Wait MPA enable 2 sample and hold	Wait defined time MPA enable to sample and hold – ld2sh_dly (reg7[5:4])
8	Write reg1 0x1093	Set bb_sample_en

9	Sampling running	Wait defined on-time – dc_on_pulse_len (reg7[9:8])
10	Write reg1 0x1092	Reset bb_sample_en Wait 100 ns
11	Write reg0 0x393F	Reset pll_active, pll_clk_gate_en Wait 100 ns
12	Write reg0 0x391F	Reset mpa_en Wait 100 ns
13	Write reg1 0x0092	Reset div_bias_on
14	Write reg0 0x0900	RF off, only pll_en ist still on and pll_open_loop is set
15	Wait for next wakeup	Sequence is restarted after defined time – dc_rep_rate (reg7[11:10]), dc_rep_rate measures from active phase to active phase

Finally, the rising edge on spi_cs_n_i indicates the end of the access.

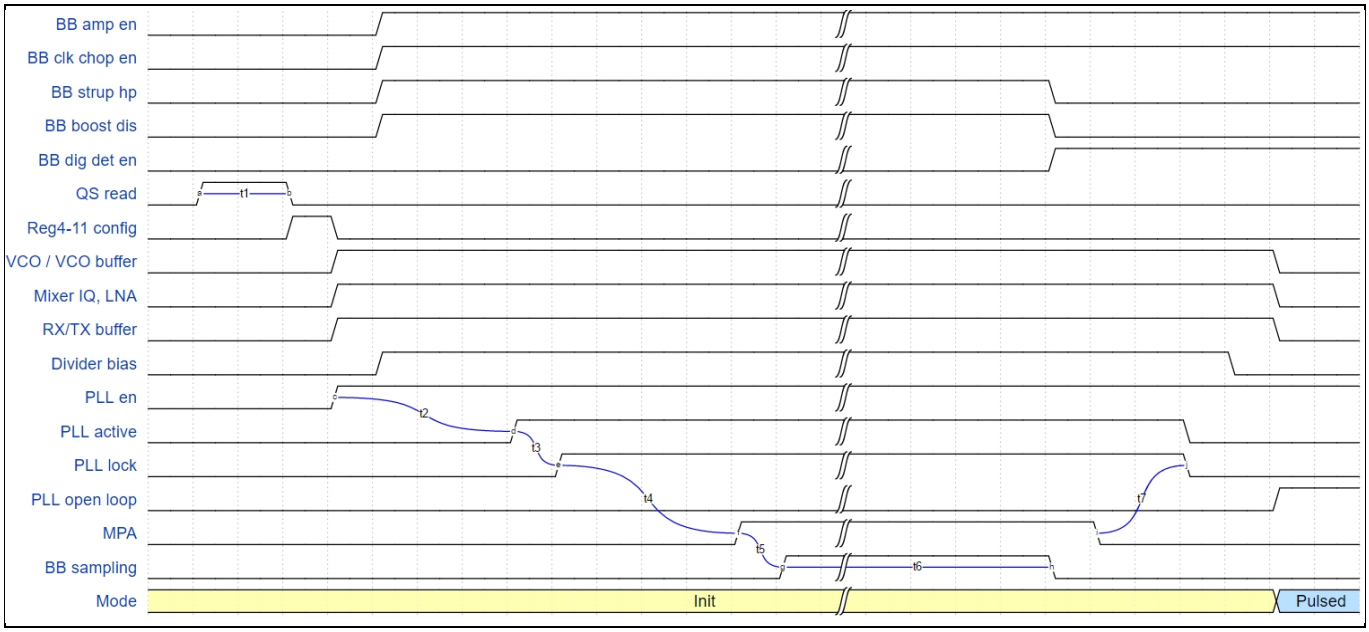

Figure 5 Pulsed mode – init phase

Figure 5 shows the wave diagram of the init phase in pulsed mode for all control signals.

t_1: 200 µs, time quad-state pin status need to be ready before reading them.

Before switching on any block the e-fuses are read and config. registers reg4–11 are written.

t_2: 2 µs, time PLL enable needs to be enabled before PLL is set into active state. Shortly after switching on of the VCO, divider bias has to be activated

t_3: ~15 µs, time PLL needs for locking

t_4: 20 µs, time needed for loop filter settling with MPA disabled

t_5: mpa2sh_dly (1 µs default), time between activating MPA and sampling

t_6: 5 ms, time needed for BB settling

t_7: 20 µs, time needed for PLL settling with MPA disabled to allow best relocking condition in pulsed mode

Now the digital detector is switched on, it takes another 50 ms until it starts counting to allow settling of the analog detector. BB is configured for pulsing mode at the same time.

PLL and RF are switched off in the order: sample and hold/MPA/PLL/divider bias/RF, each step has to take 100 ns, after MPA t_7 is needed. The signal pll_clk_gate_en has to be the same as pll_active.

Registers can be programmed externally during active pulsed mode, but the following points have to be taken into account:

- An access halts the sequencer of the main controller as it waits for the register change. BB_clk_chop, digital detector and wakeup counter are still running.
- PLL registers (reg4–6) must not be changed when pll_en = 1 (reg0[8]) or pll_clk_gate_en = 1 (reg0[10]).
- Digital detector settings (reg10, reg11) must not be changed when bb_dig_det_en = 1 (reg1[7]).
- The wakeup pulse from wakeup logic to sequencer is 1.6 µs long. If an SPI access covers the complete pulse the next power-up phase is skipped. The next wakeup will happen with the next wakeup pulse.

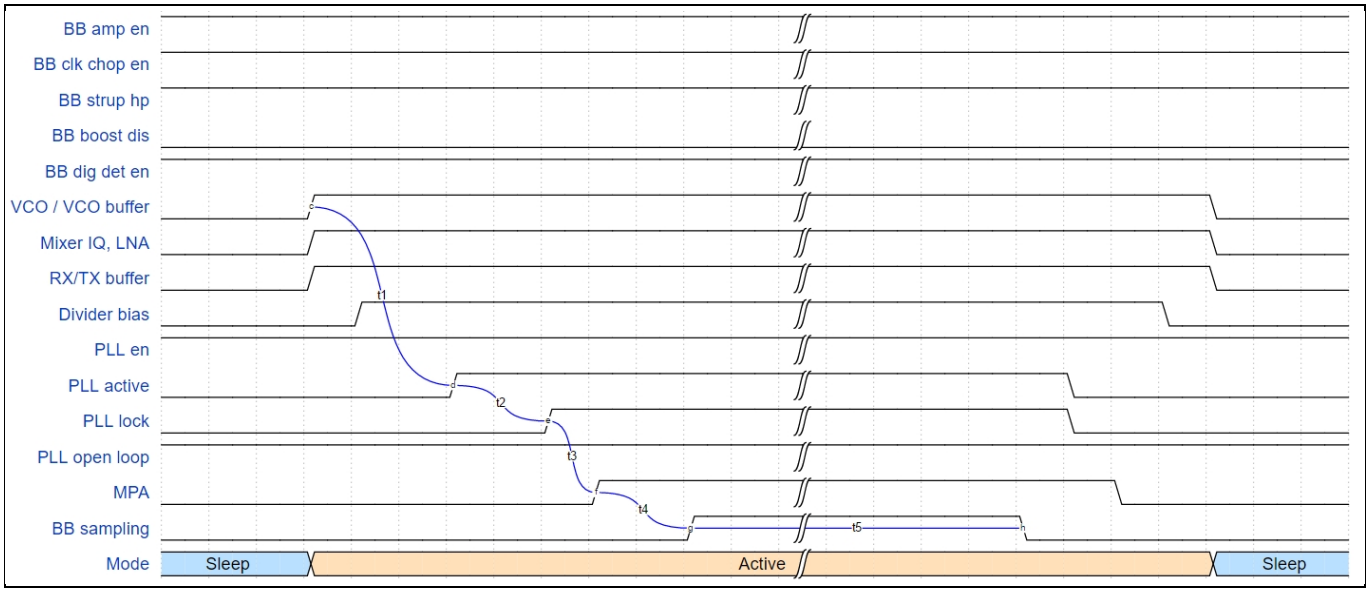

Figure 6 Pulsed mode – active phase

Figure 6 shows the pulsed mode after the init phase starting with a sleep phase and also ending with a sleep phase. Dc_rep_rate reg7[11:10] defines the time from start of active phase until start of next active phase.

t_1: vco2pll_dly (1 µs default), time PLL active is set after enabling VCO. Shortly after switching on of the VCO, divider bias has to be activated

t_2: ~5 µs, time PLL needs for locking

t_3: 300 to 400 ns, time needed for synchronizing lock detect signal and enabling MPA

t_4: mpa2sh_dly (1 µs default), time between activating MPA and sampling

t_5: dc_on_pulse_len (5 µs default) reg7[9:8], sampling on-time

PLL and RF are switched off in the order: sample and hold/PLL/MPA/divider bias/RF, each step has to take 100 ns. The signal pll_clk_gate_en has to be the same as pll_active.

1.2.4 CW sequence

In CW mode the device is active as configured with e-fuses and quad-state inputs. The main controller is inactive. The init sequence was left at the right step so the CW sequence itself contains only two entries, as only one control signal has to be switched off and the digital part of the detector has to be switched on.

Table 8 **CW sequence in detail**

No.	Command	Description
1	Write reg1 0x10B3	Set bb_dig_det_en, reset bb_strup_hp
2	End	

Figure 7 shows CW mode including init phase. Init phase until end settling time for BB is the same as for pulsed mode. As in CW mode BB and sampling is on continuously, and bb_strup_hp is set to 0. As in CW mode BB and sampling is on continuously, and bb_strup_hp is set to 0.

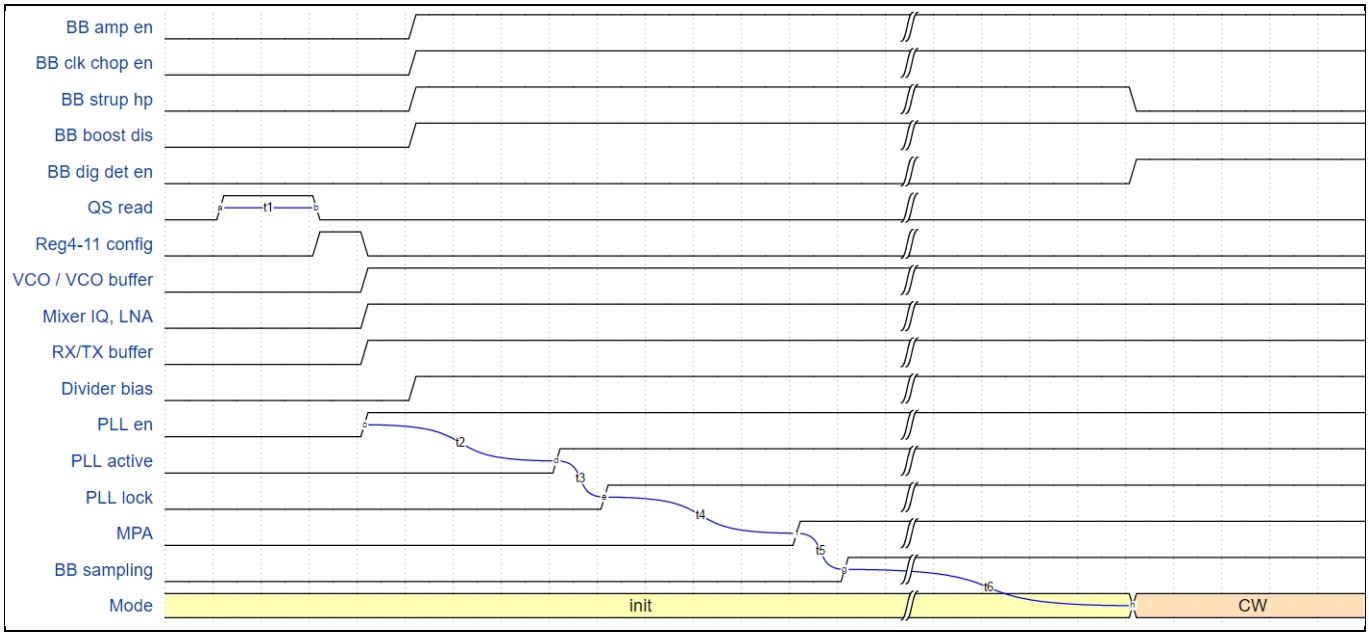

Figure 7 CW mode

1.2.5 SPI mode sequence

This is the manual mode; the main controller is inactive and reg0/1 are set to all off by the main controller. The SPI sequence just has to switch off quad-state inputs. Depending on quad-state input 1, the external 9.6 MHz clock is enabled (QS1 = "10") or not (QS1 = "11").

Table 9 **SPI mode sequence in detail**

No.	Command	Description
1	Write reg1 0x0000	Reset bit qs_rd_en
2	End	

Optionally it is also possible to configure the registers and activate the pulsed mode afterward. This can be done by setting start_pm (reg15[14]) to 1. If external clock is enabled (QS1= "10") it can be switched off with the

same SPI access by setting bit clk_ext_dis (reg15[13]). When the clock is switched off, 16 to 32 further clocks are delivered to allow the external component to enter sleep state.

1.2.6 Detector

When RF blocks are switched on, the detector can be activated. It takes another 50 ms after enabling of the detector for settling of BB. During this time the output P_{det} is kept in an inactive state.

1.3 Overview of dynamic control signals

Table 10 **Overview of dynamic control signals**

Signal name	Pulsed mode	CW mode	SPI mode
vcobuf_en, vco_en	Toggling	1	0
pll_clk_gate_en, pll_active	Toggling	1	0
pll_open_loop	0 (init phase)/ 1 (pulsed)	0	0
pll_en	1	1	0
mpa_en	Toggling	1	0
rx/txbuf_en	Toggling	1	0
mixi_en, mixq_en	Toggling	1	0
lna_en	Toggling	1	0
div_bias_en	Toggling	1	0
qs_rd_en	0	0	0
bb_dig_det_en	1	1	0
bb_boost_dis	0	1	0
bb_clk_chop_en	1	1	0
bb_strup_hp	0	0	0
bb_amp_en	1	1	0
bb_sample_en	Toggling	1	0

Dynamic control signals are used to switch on/off analog and also digital blocks. They are located in the direct access registers (=reg0/1). The main controller is able to do sequencing with full clock speed so it takes 100 ns for each register access. For all other registers the main controller has to use the SPI, which takes about 25 clock cycles (=2.5 µs).

Figure 12 shows a block diagram of this concept. When programming these bits manually, the XOR logic has to be taken into account.

If the main controller has switched off a bit it can be activated by programming it with "1". If the main controller has switched on a bit it can be activated by programming it with "0".

For clarification, here are some examples:

- If bb_clk_chop_en should be switched off, it has to be programmed for pulsed mode and CW mode with "1" and in SPI mode with "0".
- If bb_boost_dis should be switched on, it has to be programmed for pulsed mode and SPI mode with "1" and in CW mode with "0".
- If lna_en should be switched off, it should be programmed for CW mode with "1" and for SPI mode with "0". It cannot be switched off in pulsed mode as it is programmed by the main controller continuously. Programming it with "1" would just invert the bit.

2 SPI

- 7-bit continuous address space
- Fixed payload of 16 bits
- Chip-select (slave-select) active in low state

2.1 SPI description

The SPI command is read via the data input spi_di_i (serial data in), which is synchronized with the clock input spi_clk_i provided by the master (FPGA). The output word appears synchronously at the data output spi_do_o (serial data out). The transmission cycle begins when the chip is selected by the input signal spi_cs_n_i (chip select not), low active. With the last rising edge of spi_clk_i data is written into the register block. The transmission cycle ends with a rising edge on the input signal spi_cs_n_i.

The working edge is the rising edge of the clock. The status of spi_di_i is shifted into the input register with every working edge. And with every working edge the state of the spi_do_o bus is shifted out of the output.

2.1.1 SPI write mode

A write access starts after the falling edge of spi_cs_n_i with transfer of the 7-bit address, MSB first. The following 8-bit (RW = read/write bit) is "1", indicating a write access. After that the 16-bit payload is sent, also MSB first.

At the same time, while the address and RW bit are received, the global status register GSR0 (8-bit) is serially shifted out on spi_do_o (also MSB first). During sending of the payload, the previous register content is serially shifted out on spi_do_o.

Finally, the rising edge on spi_cs_n_i indicates the end of the access.

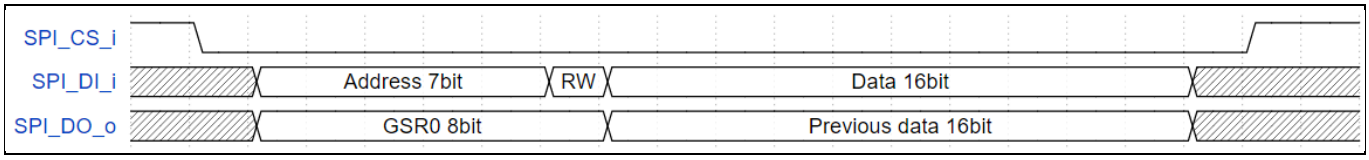

Figure 8 SPI write mode – MSB first any time, RW = "1"

2.1.2 SPI read mode

A read access starts after the falling edge of spi_cs_n_i with transfer of the 7-bit address, MSB first. The following 8-bit (RW = read/write bit) is "0", indicating a read access. The following 16-bit data are ignored as they are not needed for a read access.

At the same time, while address and RW bit are received, the global status register GSR0 (8-bit) is serially shifted out on spi_do_o (also MSB first). Directly after that the read data is serially shifted out on spi_do_o.

Finally, the rising edge on spi_cs_n_i indicates the end of the access.

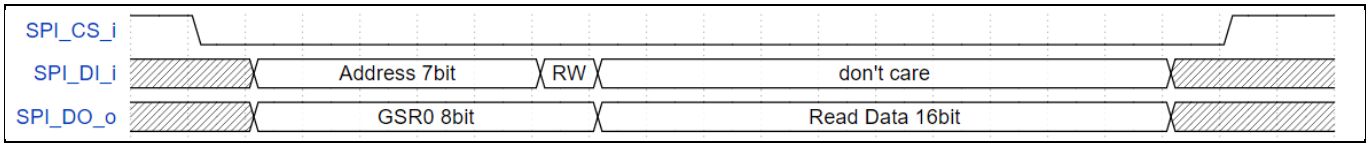

Figure 9 SPI read mode – MSB first any time, RW = "0"

2.1.3 SPI burst mode

The burst mode can be used to read or write out several registers instead of reading just single registers. The burst mode command consists of several bitfields and is shown in Table 11.

Burst command examples:

Burst command for start read from register 4: 0xFF08

Burst command for start write from register 7: 0xFF0F

Table 11 **Burst mode command**

Bitfield	Bit width	Bitfield name	Description
15:9	7	addr	Address to request burst mode = 0x7F
8	1	bit8	Bit to fill the 16-bit command = 1
7:1	7	saddr	Burst mode starting address
0	1	rwb	Burst mode read/write 0 – burst read 1 – burst write

2.1.3.1 SPI burst access

After the start condition the 16-bit burst mode command is sent from the SPI master on spi_di_i. At the same time, the status register GSR0 (8-bit) and 8-bit dummy data are shifted out on spi_do_o. After the command sequence is done, in burst write mode, the write burst data are shifted in from the SPI master on spi_di_i or the read burst data are shifted out to the SPI master on spi_do_o in burst read mode.

For burst accesses, any number of written/read data blocks can be used. The access is ended by a rising edge of spi_cs_i.

Burst mode read sequence:

In the burst mode read sequence, the SPI master reads from the device.

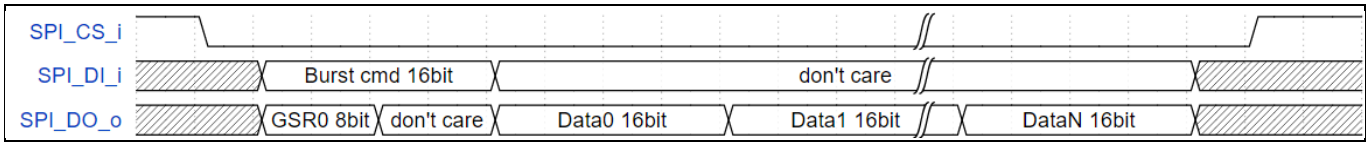

Figure 10 SPI burst read

Burst mode write sequence:

In the burst write mode, the SPI master writes to the device.

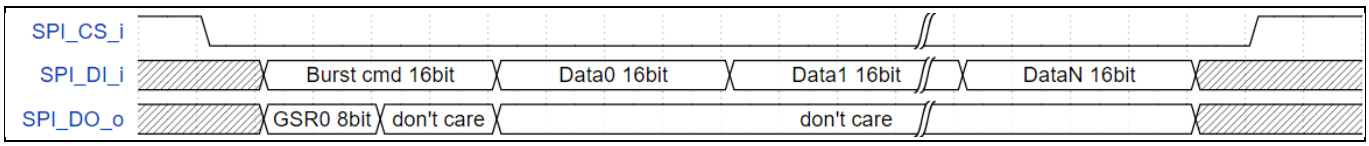

Figure 11 SPI burst write

2.2 SPI register

2.2.1 Register overview

Table 12 **Register overview including reset values for e-fuse = 0 x0000**

Register	Mode	Contents	Reset value	Value after init sequence (pulsed sleep/CW)
reg0	RW	Control bits	0x0000	0x0900*/0x373F*
reg1	RW	Control bits	0x0000	0x0092*/0x10B3*
reg2	RW	Reserved	0x0000	0x0000
reg3	RW	Reserved	0x0000	0x0000
reg4	RW	PLL 1	0x0000	0x053A
reg5	RW	PLL 2	0x0000	Dep. on QS4
reg6	RW	PLL 3	0x0000	0x6800
reg7	RW	Duty cycling, timing, PD, MPA	0x0000	0x0457
reg8	RW	Divider	0x0000	0x0000
reg9	RW	BB	0x0000	0x0068/0x0078
reg10	RW	QS inputs	0x0000	Dep. on QS2/3
reg11	RW	BB detector	0x8000	0x0018
reg12	RW	BITE, amux	0x0000	0x0000
reg13	RW	Reserved	0x0000	0x0000
reg14	RW	Reserved	0x0000	0x0000
reg15	RW	Digital control	0x0000	0x0000
reg34	RW	ADC start	0x0000	0x0000
reg35	RW	ADC convert	0x0000	0x0000
reg36	RO	ADC status	0x0000	0x0000
reg38–53	RO	ADC result channel 0 to 15	n/a	n/a
reg55	RO	E-fuse	0x8019	0x0000
reg56	RO	Status and chip version	0x0001	0x2001, bit(2:0) dep. on chip version Bit(15:14) dep. on QS1
GSR0	RO	8-bit SPI status register	0x00	0x00

*These values are set by the main controller, therefore a register read will deliver 0x0000.

For the reset values a distinction is necessary between reset value directly after reset and reset value when reading the values is possible. The reason for this is that the reset values are overwritten by the init sequence for pulsed and CW mode. The "real" reset values can be read only in SPI mode, as they are not changed in this mode.

2.2.2 Direct access register

reg0 and reg1 are direct access registers shown in Figure 12. These bits can be controlled directly by the main controller. All other registers needs to be programmed by the SPI from the external optional microcontroller or internally from the main controller at power-up.
As an external read access from SPI any time addresses the register within SPI ADC block the information on the output of the XOR cannot be read. A read access delivers the value stored within the SPI ADC.

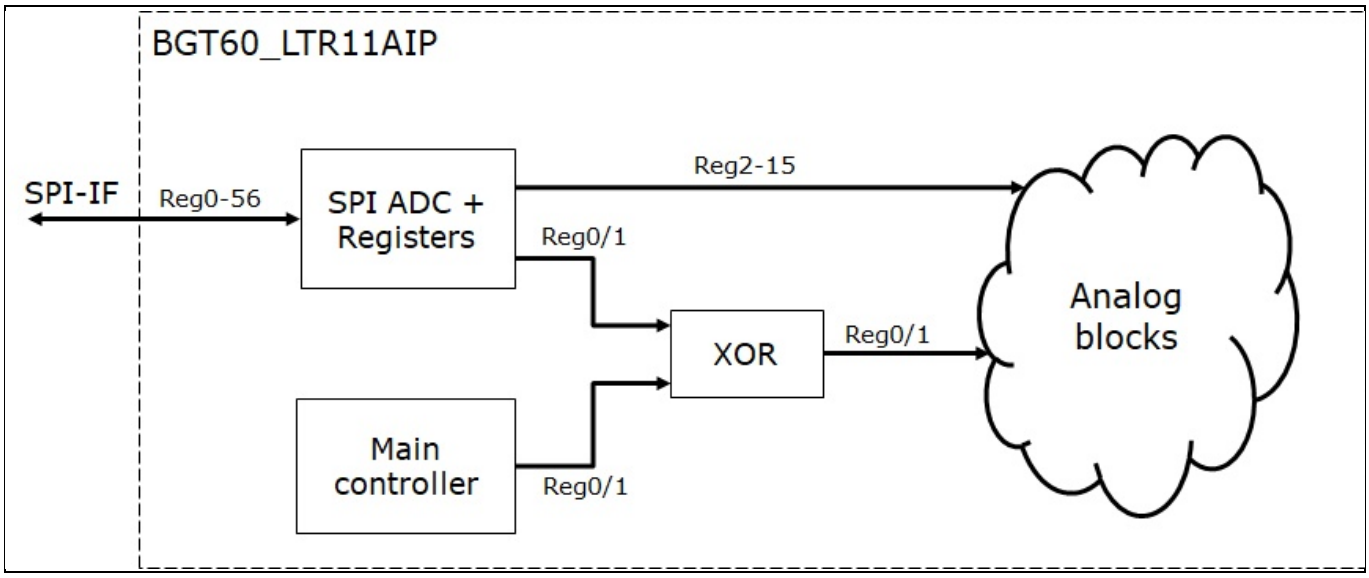

Figure 12 Direct access registers

Table 13 **XOR**

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

2.2.3 Register map bitfields

Table 14 **Register map**

		7	6	5	4	3	2	1	0
reg0	[15:8]			vcobuf_en	vco_en	pll_open_loop	pll_clk_gate_en	pll_active	pll_en
	[7:0]			mpa_en	txbuf_en	mixq_en	mixi_en	lna_en	rxbuf_en
reg1	[15:8]				div_bias_en				qs_rd_en
	[7:0]	bb_dig_det_en		bb_boost_dis	bb_clk_chop_en		bb_strup_hp	bb_amp_en	bb_sample_en
reg2	[15:8]								
	[7:0]								

		7	6	5	4	3	2	1	0
reg3	[15:8]								
	[7:0]								
reg4	[15:8]	pll_dft_dmux			pll_bias_dis	pll_lf_iso	pll_lf_r4_sel	pll_cl_loop_pmode	pll_lf_r2_sel
	[7:0]	xosc_mode	pll_fbdiv_cnt	pll_cp_icp_sel[2:0]			pll_cp_mode	pll_pfd_rdt_sel[1:0]	
reg5	[15:8]					pll_fcw[11:8]			
	[7:0]	pll_fcw[7:0]							
reg6	[15:8]	pll_ld_tw_sel[2:0]			pll_ld_len	pll_ld_en			
	[7:0]								
reg7	[15:8]					dc_rep_rate[1:0]		dc_on_pulse_len[1:0]	
	[7:0]		vco2pll_dly	mpa2sh_dly[1:0]		pd_en	mpa_ctrl[2:0]		
reg8	[15:8]								
	[7:0]					div_sel[1:0]		div_out_en	div_test mode_en
reg9	[15:8]								
	[7:0]	bb_hp_res[1:0]		bb_clk_chop_sel	bb_lpf_bw	bb_ctrl_gain[3:0]			
reg10	[15:8]					bb_det_hold[1:0]		bb_det_hold_mul[1:0]	
	[7:0]	bb_det_thrs[1:0]		bb_det_thrs_fine[1:0]					
reg11	[15:8]								
	[7:0]				bb_det_dir_divcnt	bb_det_cnt_reset[1:0]		bb_det_cnt[1:0]	
reg12	[15:8]								
	[7:0]	bb_amux_ctrl		bb_amux_en	bite_pd_en	bite_ctrl[2:0]			bite_en
reg13	[15:8]								
	[7:0]								
reg14	[15:8]								
	[7:0]								
reg15	[15:8]	soft_reset	start_pm	clk_ext_dis					
	[7:0]								
reg34	[15:8]	reserved							
	[7:0]	reserved					adc_en	bandgap_en	adc_clk_en
reg35	[15:8]	reserved							

		7	6	5	4	3	2	1	0
	[7:0]	lv_gain	reserved		channel_all	channel_nr			
reg36	[15:8]	reserved							
	[7:0]	reserved						adc_ready	bg_up
reg38 –53	[15:0]	ADC result register channel 0 to 15							
reg55	[15:8]	fuse_out							
	[7:0]								
reg56	[15:8]	quad_state1		init_done					
	[7:0]				pll_pa_ active	pll_lock	Chip_version		

2.2.4 Register reg0 – direct access register

Table 15 **Register assignment of reg0**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
		vcobuf_en	vco_en	pll_open_loop	pll_clk_gate_en	pll_active	pll_en			mpa_en	txbuf_en	mixq_en	mixi_en	lna_en	rxbuf_en

Reset value: 0x0000
Value after init sequence: 0x0900 for pulsed mode
Value after init sequence: 0x373F for CW mode

Table 16 **Signal table of reg0**

Signal name	Size	Function
vcobuf_en	1	**Enable VCO buffer** 0_D: VCO buffer off 1_D: VCO buffer on
vco_en	1	**Enable VCO** 0_D: VCO off 1_D: VCO on

Signal name	Size	Function
pll_open_loop	1	**PLL open-loop clock gating enable** Switches PLL into open loop after lock detect was reached. 0_D: closed loop 1_D: open loop after lock detect
pll_clk_gate_en	1	**PLL clock gating enable** Activates clock for digital portion of the PLL. Synchronized within PLL. 0_D: PLL dig. clock off 1_D: PLL dig. clock on
pll_active	1	**PLL active** PLL locking is started when this bit is set. 0_D: PLL loop open 1_D: PLL locking started
pll_en	1	**Enable PLL** This bit enables bias structures of the PLL. PLL config. register must be stable as long as pll_en is 1. 0_D: PLL disabled 1_D: PLL enabled
mpa_en	1	**Medium power amplifier enable** 0_D: MPA off 1_D: MPA on
txbuf_en	1	**Enable TX buffer** 0_D: TX buffer off 1_D: TX buffer on
mixq_en	1	**Enable mixer Q** 0_D: Mixer Q off 1_D: Mixer Q on
mixi_en	1	**Enable mixer I** 0_D: Mixer I off 1_D: Mixer I on
lna_en	1	**Enable LNA** 0_D: LNA off 1_D: LNA on
rxbuf_en	1	**Enable RX buffer** 0_D: RX buffer off 1_D: RX buffer on

2.2.5 Register reg1 – direct access register

Table 17 **Register assignment of reg1**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
			div_bias_en				qs_rd_en	bb_dig_det_en		bb_boost_dis	bb_clk_chop_en		bb_strup_hp	bb_amp_en	bb_sample_en

Reset value: 0x0000
Value after init sequence: 0x0092 for pulsed mode
Value after init sequence: 0x10B3 for CW mode

Table 18 **Signal table of reg1**

Signal name	Size	Function
div_bias_en	1	**Enable divider bias** 0_D: Divider bias off 1_D: Divider bias on
qs_rd_en	1	**Enable quad-state input** The quad-state inputs have to be enabled 200 µs before reading of the inputs to allow analog settling. 0_D: QS off 1_D: QS on
bb_dig_det_en	1	**Enable digital BB detector** Enables digital part of detector. After first switching on of this bit after start-up/chip reset it takes 50 ms until the digital part of the detector starts counting target hits to allow settling of analog circuit. 0_D: BB detector off 1_D: BB detector on
bb_boost_dis	1	**BB sample and hold switch boost setting** 0_D: Sample and hold gate voltage boost is enabled (pulsed mode) 1_D: Sample and hold gate voltage boost is disabled (CW mode)
bb_clk_chop_en	1	**Enable clock chop** This bit enables continuous clock signal for chopping. 0_D: Clock off 1_D: Clock on
bb_strup_hp	1	**BB start-up boost mode** 0_D: Start-up boost mode disabled 1_D: Start-up boost mode enabled

Signal name	Size	Function
bb_amp_en	1	**Enable BB amplifier** 0_D: BB amplifier disabled 1_D: BB amplifier enabled
bb_sample_en	1	**Enable BB sampling** Controls connection of mixer output to sample and hold capacitance. 0D: Disconnected, hold phase 1D: Connected, sampling phase

2.2.6 Register reg4 – PLL config 1

Table 19 **Register assignment of reg4**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
pll_dft_dmux			pll_bias_dis	pll_lf_iso	pll_lf_r4_sel	pll_cl_loop_pmode	pll_lf_r2_sel	xosc_mode	pll_fbdiv_cnt	pll_cp_icp_sel			pll_cp_mode	pll_pfd_rdt_sel	

Reset value: 0x0000
Value after init sequence (unfused): 0x053 A

This register must not be changed when pll_en = 1 (reg0[8]).

Table 20 **Signal table of reg4**

Signal name	Size	Function
pll_dft_mux	2	**PLL data mux for DFT** Default after init sequence: 0_D The chip output SPI_DO is used to make the PLL test information visible outside. Of course SPI read accesses will not work when this bitfield is not set to functional mode, but SPI write accesses will still work. PLL lock is the internal PLL lock, not the one connected to the digital state machine. It is without the delay, which can be configured by the bitfield pll_ld_len. The internal PLL lock signal is permanently 1 if lock detection is disabled. 0_D: Functional mode 1_D: PLL lock 2_D: Reference clock divided by 4 3_D: Divider clock divided by 4

Signal name	Size	Function
pll_bias_dis	1	**PLL bias disable** Default after init sequence: 0_D Disables bandgap in PLL and V^2I converter (=PLL biasing). Can be set to further reduce current consumption in SPI mode. Also disables clock for internal main controller, therefore pulsing is not possible if PLL biasing is switched off. 0_D: Biasing on 1_D: Biasing off
pll_lf_iso	1	**Loop filter isolation mode** Default after init sequence: 0_D 0_D: Isolation with switches only 1_D: Isolation with charge-keeping buffer enabled
pll_lf_r4_sel	1	**Loop filter R4 setting** Default after init sequence: $1_D/0_D$ E-fuse used 0_D: 12.4 k 1_D: 0.1 k
pll_cl_loop_pmode	1	**Closed loop in pulsed mode** Default after init sequence: 0_D pll_open_loop reg0[11] controls open/closed loop of the PLL after lock of the PLL. This bit is set by the main controller in pulsed mode. By setting pll_cl_loop_pmode open loop is also used for pulsed mode. 0_D: Closed loop mode used in pulsed mode (pll_open_loop forced to 0) 1_D: Open loop mode used in pulsed mode
pll_lf_r2_sel	1	**Loop filter R2 setting** Default after init sequence: $1_D/0_D$ E-fuse used 0_D: 21.6 k 1_D: 18.7 k
xosc_mode	1	**XTAL oscillator mode** Default after init sequence: 0_D 0_D: Amplitude setting 1 1_D: Amplitude setting 2
pll_fbdiv_cnt	1	**Feedback divider counter setting** Default after init sequence: 0_D 0_D: 60 GHz – mode cntA = 21 dB for 38.4 MHz 1_D: 60 GHz – mode cntB = 20 dB for 40 MHz

Signal name	Size	Function
pll_cp_icp_sel	3	**Charge pump current setting** Default after init sequence: 55 µA/30 µA (7_D/2_D) E-fuse used 0_D: 20 µA 1_D: 25 µA 2_D: 30 µA 3_D: 35 µA 4_D: 40 µA 5_D: 45 µA 6_D: 50 µA 7_D: 55 µA
pll_cp_mode	1	**Charge pump bias mode** Default after init sequence: 0_D/1_D E-fuse used 0_D: Bias regulation loop active 1_D: Fix bias mode = bias regulation loop off
pll_pfd_rdt_sel	2	**PFD reset delay time select** Default after init sequence: 2_D 0_D: 175 ps 1_D: 275 ps 2_D: 375 ps 3_D: 470 ps

2.2.7 Register reg5 – PLL config 2

Table 21 **Register assignment of reg5**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
				pll_fcw											

Reset value: 0x0000
Value after init sequence: dependent on QS4 and pll_japan_mode e-fuse

This register must not be changed when pll_en = 1 (reg0[8]).

Table 22 **Signal table of reg5**

Signal name	Size	Function
pll_fcw	12	**PLL frequency word** Default after init sequence: Dependent on QS4 and pll_japan_mode e-fuse FCW for f_{start} (4-bit integer + 8-bit fractional) → 2.4 MHz raster at 60 GHz Predefined settings for Japan mode: 60.6 GHz: 0xEA2 60.7 GHz: 0xECC 60.8 GHz: 0xEF5 60.9 GHz: 0xF1F Predefined settings for Europe mode: 61.1 GHz: 0xF72 61.2 GHz: 0xF9C 61.3 GHz: 0xFC6 61.4 GHz: 0xFEF

2.2.8 Register reg6 – PLL config 3

Table 23 **Register assignment of reg6**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
pll_ld_tw_sel			pll_ld_len	pll_ld_en											

Reset value: 0x0000
Value after init sequence (unfused): 0x6800

This register must not be changed when pll_en = 1 (reg0[8]).

Table 24 **Signal table of reg6**

Signal name	Size	Function
pll_ld_tw_sel	3	**Lock detection time window** Default after init sequence: $3_D/2_D/4_D/5_D$ E-fuse used Accepted phase difference for lock detection condition within comparator (typical values). 0_D: 0.26 ns 1_D: 0.5 ns 2_D: 1.0 ns 3_D: 1.5 ns 4_D: 2.0 ns 5_D: 2.8 ns 6_D: 3.8 ns 7_D: 4.6 ns
pll_ld_len	1	**Lock detection – lock assertion condition + lock detection – lock delay** Default after init sequence: $0_D/1_D$ E-fuse used This bit has two functions. Lock assertion condition: Number of consecutive clock cycles; the lock criteria must be fulfilled 0_D: 24 clock cycles 1_D: 16 clock cycles Lock detection delay time t_{delay_lock}: Time between lock detection and rising edge on lock detect signal 0_D: 3.57 µs 1_D: 5.23 µs
pll_ld_en	1	**Enable lock detection** Default after init sequence: 1_D 0_D: Lock detection off + lock bit forced to high after t_{delay_lock} when PLL is active. t_{delay_lock} is programmable by pll_ld_len. 1_D: Lock detection on

2.2.9 Register reg7 – duty cycling, timing, PD, MPA

Table 25 **Register assignment of reg7**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
				dc_rep_rate		dc_on_pulse_len			vco2pll_dly	mpa2sh_dly		pd_en	mpa_ctrl		

Reset value: 0x0000
Value after init sequence: 0x0457

Table 26 **Signal table of reg7**

Signal name	Size	Function
dc_rep_rate	2	**Duty cycle repetition rate** Default after init sequence: $1_D/2_D$ E-fuse used Defines the time until next pulsing sequence starts in pulsing mode. 0_D: 250 µs (10 km/h) 1_D: 500 µs 2_D: 1000 µs 3_D: 2000 µs
dc_on_pulse_len	2	**Duty cycle on pulse length** Default after init sequence: $0_D/1_D$ E-fuse used Defines the time sampling is active during one pulsing event. 0_D: 5 µs 1_D: 10 µs 2_D: 20 µs 3_D: 40 µs
vco2pll_dly	1	**VCO to PLL delay** Default after init sequence: 1_D Defines the time PLL is enabled after VCO is enabled. 0_D: 500 ns 1_D: 1000 ns
mpa2sh_dly	2	**MPA enable to sample and hold delay** Default after init sequence: 1_D Defines the time sample and hold is activated after PLL lock was reached and MPA was enabled. 0_D: 500 ns 1_D: 1000 ns 2_D: 2000 ns 3_D: 4000 ns
pd_en	1	**Enable PD** Default after init sequence: 0_D 0_D: PD off 1_D: PD on

Signal name	Size	Function
mpa_ctrl	3	**Medium power amplifier gain control** Default after init sequence: $7_D/5_D/3_D/1_D$ E-fuse used Preliminary values, to be checked in lab. 0_D: -34 dBm 1_D: -31.5 dBm 2_D: -25 dBm 3_D: -18 dBm 4_D: -11 dBm 5_D: -5 dBm 6_D: 0 dBm 7_D: 4.5 dBm

2.2.10 Register reg8 – divider

Table 27 **Register assignment of reg8**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
												div_sel		div_out_en	div_testmode_en

Reset value: 0x0000
Value after init sequence: 0x0000

Table 28 **Signal table of reg8**

Signal name	Size	Function
div_sel	2	**Divider select** Default: 0_D Selects frequency divider setting. In default state internal 9.6 MHz clock is selected. This clock is active only if SPI mode with external clock is selected by QS1 input and the disable bit clk_ext_dis (reg15[13]) is not set. 0_D: Select internal 9.6 MHz clock from oscillator 1_D: 2^14 2_D: 2^17 3_D: 2^21

Signal name	Size	Function
div_out_en	1	**Divider out enable** Default: 0_D Enables the 2^14, 2^17, 2^21 divider logic. Does not affect setting 0 from div_sel, internal clock on pad div_out is enabled, if corresponding mode is selected with QS1 and test mode is off (div_testmode_en = 0). 0_D: Divider out off 1_D: Divider out on
div_testmode_en	1	**Enable divider test mode** Default: 0_D Puts VCO frequency divided by 16 on pad div_out. Overrides setting from bitfield div_sel. 0_D: Test mode off (div_sel active) 1_D: Test mode on

2.2.11 Register reg9 – BB

Table 29 **Register assignment of reg9**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								bb_hp_res		bb_clk_chop_sel	bb_lpf_bw	bb_ctrl_gain			

Reset value: 0x0000
Value after init sequence: 0x0066 for pulsed mode
Value after init sequence: 0x0076 for CW mode

Table 30 **Signal table of reg9**

Signal name	Size	Function
bb_hp_res	2	**High-pass filter resistor settings** Default: $1_D/3_D$ E-fuse used 00_B: 8 MΩ 01_B: 4 MΩ 10_B: 2 MΩ 11_B: 1 MΩ

Signal name	Size	Function
bb_clk_chop_sel	1	**Select clock chop frequency** Default: 1_D Selects frequency of clock for chopping (input for analog). 0_D: 100 kHz 1_D: 200 kHz
bb_lpf_bw	1	**Low-pass filter setting** Default: 0_D 0_D: 10 kHz (pulsed mode) 1_D: 60 kHz (CW mode)
bb_ctrl_gain	4	**BB PGA gain setting** Default: 40 dB/10 dB/25 dB/50 dB ($6_D/0_D/3_D/8_D$) E-fuse used 0_D: 10 dB 1_D: 15 dB 2_D: 20 dB 3_D: 25 dB 4_D: 30 dB 5_D: 35 dB 6_D: 40 dB 7_D: 45 dB 8_D: 50 dB

2.2.12 Register reg10 – quad-state inputs

Table 31 **Register assignment of reg10**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
reserved				bb_det_hold		bb_det_hold_mul		bb_det_thrs		bb_det_thrs_fine		reserved			

Reset value: 0x0000
Value after init sequence: Dependent on QS2 and QS3, 0x0010 for QS2 = QS3 = 0

The bitfields bb_det_hold and bb_det_hold_mul must not be changed when bb_dig_det_en = 1 (reg1[7]).

Table 32 **Signal table of reg10**

Signal name	Size	Function
bb_det_hold	2	**BB detector holdtime** Register is overwritten with content of QS3 (quad-state input) at start-up. Sets holdtime of chip output target_det. It defines the time between last target detector counter reaching defined hit count to falling edge of target_det. 00_B: 10 ms 01_B: 1 s 10_B: 10 s 11_B: 1 min.
bb_det_hold_mul	2	**BB detector holdtime multiplicator** Default after init sequence: 0_D Additional possibility to adapt holdtime. $t_{holdtime}$ = bb_det_hold * bb_det_hold_mul 00_B: 1 01_B: 2 10_B: 4 11_B: 8
bb_det_thrs	2	**BB comparator threshold** Register is overwritten with content of QS2 (quad-state input) at start-up. Sets course voltage when detector comparator will switch. 00_B: 787.5 mV 01_B: 937.5 mV 10_B: 1087.5 mV 11_B: 1237.5 mV Used voltage at comparator = voltage set by bb_det_thrs + voltage set by bb_det_thrs_fine.
bb_det_thrs_fine	2	**BB comparator threshold fine-tuning** Default after init sequence: 1_D Fine-tuning of voltage level when detector comparator will switch. 01_B: 0 mV 00_B: 37.5 mV 11_B: 75 mV 10_B: 112.5 mV Used voltage at comparator = voltage set by bb_det_thrs + voltage set by bb_det_thrs_fine

2.2.13 Register reg11 – BB detector

Table 33 **Register assignment of reg11**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
											bb_det_dir_divcnt	bb_det_cnt_reset		bb_det_cnt	

Reset value: 0x8000
Value after init sequence (unfused): 0x001B

This register must not be changed when bb_dig_det_en = 1 (reg1[7]).

Table 34 **Signal table of reg11**

Signal name	Size	Function
bb_det_dir_divcnt	1	**BB detector direction counter hit setting** Default after init sequence: 1_D Defines number of hits needed to set the chip output "target_dir". When target on any of the two direction counters is reached, both counters are reset and "target_dir" is written. 0_B: Reset at bb_det_cnt 1_B: Reset at bb_det_cnt/2
bb_det_cnt_reset	2	**BB detector hit counter reset setting** Default after init sequence: 100 ms/10 ms (2_D/0_D) E-fuse used Defines a time to detect that a target is gone. If this time elapses before the next hit is detected, detector counter and both direction counters are reset. 0_D: 10 ms (100 Hz) – reject 50 Hz interferer 1_D: 50 ms 2_D: 100 ms (10 Hz) 3_D: 200 ms
bb_det_cnt	2	**BB detector hit counter setting** Default after init sequence: 80/10 (3_D/0_D) E-fuse used Defines number of hits needed to set the chip output target_det. 0_D: 10 1_D: 20 2_D: 40 3_D: 80

2.2.14 Register reg12 – BITE

Table 35 **Register assignment of reg12**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								bb_amux_ctrl		bb_amux_en	bite_pd_en	bite_ctrl			bite_en

Reset value: 0x0000

Table 36 **Signal table of reg12**

Signal name	Size	Function
bb_amux_ctrl	2	**Selects analog voltage on QS4 pad** Default after init sequence: 0_D 0_D: BB bandgap voltage 1_D: Temperature sensor voltage 2_D: Common mode voltage I channel 3_D: Common mode voltage Q channel
bb_amux_en	1	**Enable analog voltage mux on QS4 pad** Default after init sequence: 0_D 0_B: amux off 1_B: amux on
bite_pd_en	1	**Enable BITE power detector** Default after init sequence: 0_D 0_B: BITE PD off 1_B: BITE PD on
bite_ctrl	3	**Control BITE settings** Default after init sequence: 0_D Controls phase in degrees 0_D: 0 1_D: 45 2_D: 90 3_D: 135 4_D: 180 5_D: 225 6_D: 270 7_D: 315
bite_en	1	**Enable BITE** Default after init sequence: 0_D 0_B: BITE disabled 1_B: BITE enabled

2.2.15 Register reg15 – digital control

Table 37 **Register assignment of reg15**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
soft_reset	start_pm	clk_ext_dis													

Reset value: 0x0000
Value after init sequence: 0x0000

Table 38 **Signal table of reg15**

Signal name	Size	Function
soft_reset	1	**Soft reset** Default after init sequence: 0_D Possibility to reset all digital parts (SPI ADC, main controller, PLL dig.) asynchronously by software. Also the register itself is reset, therefore writing 0 to this bit is not necessary. 0_B: Reset inactive 1_B: Reset active
start_pm	1	**Start pulsed mode** Default after init sequence: 0_D With this bit it is possible to start the pulsed mode from SPI mode. A typical use case is to configure registers and start the pulsed mode afterward by setting this bit to 1. This is the only allowed usage of this bit. 0_B: Inactive 1_B: Rising edge triggers pulsed mode
clk_ext_dis	1	**Disable external clock** Default after init sequence: 0_D In case the external clock is switched on by selecting the SPI mode with external clock enabled, it can be switched off by setting this bit to 1. After switching off, 16 to 32 further clock edges are delivered. 0_B: Clock not disabled 1_B: Clock disabled

2.2.16 Register reg34 – ADC start

Table 39 **Register assignment of reg34**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
reserved													adc_enable	bandgap_enable	clk_enable

Reset value: 0x0000

Table 40 **Signal table of reg34**

Signal name	Size	Function
adc_enable	1	**ADC block enable** Default: 0_D 0_D: ADC disabled 1_D: ADC enabled
bandgap_enable	1	**Bandgap enable** Default: 0_D This bandgap is needed for ADC. 0_D: Bandgap disabled 1_D: Bandgap enabled
clk_enable	1	**ADC clock enable** Default: 0_D 0_D: ADC clock disabled 1_D: ADC clock enabled

2.2.17 Register reg35 – ADC convert

Table 41 **Register assignment of reg35**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
reserved								lv_gain	reserved		chnr_all	chnr			

Reset value: 0x0000

Table 42 **Signal table of reg35**

Signal name	Size	Function
lv_gain	1	**lv_gain** Default: 0_D Gain configuration for the analog input channels Recommendation: use setting of 1 to increase accuracy. 0_D: Gain = 0.75, full-scale analog input voltage 1.613 V 1_D: Gain = 1.00, full-scale analog input voltage 1.21 V
chnr_all	1	**Channel number all** Default: 0_D 0_D: chnr selects channel to convert 1_D: Converts all 16 channels, chnr is ignored
chnr	4	**Channel number** Default: 0_D Analog input channel number selected for sampling.

A write access to reg35 starts ADC conversion with the selected settings, even if the same data is written into the register.

2.2.18 Register reg36 – ADC status

Table 43 **Register assignment of reg36**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
reserved														adc_ready	bg_up

Reset value: 0x0000

Read-only

Table 44 **Signal table of reg36**

Signal name	Size	Function
adc_ready	1	**ADC ready flag** Default: 0_D This flag indicates if the ADC is ready to work. 0_D: ADC not activated or still booting 1_D: ADC ready

Signal name	Size	Function
bg_up	1	**Bandgap_up** Default: 0_D This flag indicates if the bandgap is running. 0_D: Bandgap not running or still booting 1_D: Bandgap running

2.2.19 Register reg38–53 – ADC result

Read-only

These are the result registers of the ADC: a result is 10 bits wide, bits 0 to 9 of each register are occupied. Bits 10 to 15 are not used. As the ADC is physically an 8-bit ADC also bit 0 and bit 1 are not used. Unused bits will deliver a zero when read.

Table 45 **Signal table of reg38–53**

Channel	Reg.	Function
0	38	Power sensor MPA output
1	39	Power sensor MPAx output
2	40	IFI
3	41	IFQ
4	42	Power sensor bite_pd_out
5	43	Power sensor bite_pd_outx
6	44	TX base-biasing voltage
7	45	TX tail-biasing voltage
8	46	CM voltage IFI
9	47	CM voltage IFQ
10	48	V_{DD} RF close to SPI
11	49	V_{DD} PLL
12	50	Temperature sensor
13	51	PLL bandgap voltage
14	52	ADC bandgap voltage
15	53	ABB bandgap voltage

2.2.20 Register reg55 – e-fuses

Table 46 **Register assignment of reg55**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
pll_japan_mode	pll_cp_mode	pll_cp_icp_sel	pll_lf_r4_sel	pll_ld_len	pll_ld_tw_sel		mpa_ctrl		dc_on_pulse_len	dc_rep_rate	bb_ctrl_gain		bb_hp_res	bb_det_cnt_reset	bb_det_cnt

Reset value: 0x0000

Read-only

These bits influence the default value of corresponding bitfields of registers 4 to 11.

Table 47 **Signal table of reg55**

Signal name	Size	Function
pll_japan_mode	1	**reg5[11:0]** pll_fcw – selects frequency together with setting of QS4 0_D: Japan mode = 0_D 1_D: Europe mode = 1_D
pll_cp_mode	1	**reg4[2]** 0_D: pll_cp_mode = 0_D 1_D: pll_cp_mode = 1_D
pll_cp_icp_sel	1	**reg4[5:3], reg4[8]** 0_D: pll_cp_icp_sel = 7_D pll_lf_r2_sel = 1_D 1_D: pll_cp_icp_sel = 2_D pll_lf_r2_sel = 0_D
pll_lf_r4_sel	1	**reg4[10]** 0_D: pll_lf_r4_sel = 1_D 1_D: pll_lf_r4_sel = 0_D
pll_ld_len	1	**reg6[12]** 0_D: pll_ld_len = 0_D 1_D: pll_ld_len = 1_D
pll_ld_tw_sel	2	**reg6[15:13]** 0_D: pll_ld_tw_sel = 3_D 1_D: pll_ld_tw_sel = 2_D 2_D: pll_ld_tw_sel = 4_D 3_D: pll_ld_tw_sel = 5_D
mpa_ctrl	2	**reg7[2:0]** 0_D: mpa_ctrl = 7_D 1_D: mpa_ctrl = 5_D 2_D: mpa_ctrl = 3_D 3_D: mpa_ctrl = 1_D
dc_on_pulse_len	1	**reg7[9:8]** 0_D: dc_on_pulse_len = 0_D 1_D: dc_on_pulse_len = 1_D
dc_rep_rate	1	**reg7[11:10]** 0_D: dc_rep_rate = 1_D 1_D: dc_rep_rate = 2_D
bb_ctrl_gain	2	**reg9[3:0]** 0_D: bb_ctrl_gain = 6_D 1_D: bb_ctrl_gain = 0_D 2_D: bb_ctrl_gain = 3_D 3_D: bb_ctrl_gain = 8_D

Signal name	Size	Function
bb_hp_res	1	**reg9[7:6]** 0_D: bb_hp_res = 1_D 1_D: bb_hp_res = 3_D
bb_det_cnt_reset	1	**reg11[3:2]** 0_D: bb_det_cnt_reset = 2_D 1_D: bb_det_cnt_reset = 0_D
bb_det_cnt	1	**reg11[1:0]** 0_D: bb_det_cnt = 3_D 1_D: bb_det_cnt = 0_D

2.2.21 Register reg56 – status and chip version

Table 48 **Register assignment of reg56**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
quad_state1		init_done										pll_lock_detect	chip_version		

Reset value: Depending on chip_version, all others are 0
Value after init sequence: Depending on chip_version and QS1, init_done = 1, pll_lock_detect = 1

Read-only

Table 49 **Signal table of reg56**

Signal name	Size	Function
quad_state1	2	**Quad-state input 1** These bits contain the read value from QS1 input, which is read during the initial sequence after power-up. All other QS inputs can be read indirectly by reading the bitfields influenced by them. 00_B: QS1 = 0 CW mode 01_B: QS1 = open Pulsed mode 10_B: QS1 = 100 kΩ to V_{DD} SPI mode with ext. 9.6 MHz clock enabled 11_B: QS1 = 1 SPI mode
init_done	1	**Init sequence done** Default after init sequence: 1_D This input is set as soon as the main controller has completed the init sequence. 0_D: Initial sequence not done 1_D: Initial sequence done

Signal name	Size	Function
pll_lock_detect	1	**PLL lock detect** Default after init sequence: 1_D This input comes directly from the PLL and shows if it is currently locked. 0_D: PLL not locked 1_D: PLL locked
chip_version	3	**Chip version** Default: Sample dependent Every variant has its own version number; it is hardwired on the analog top level. These bits are read-only.

2.2.22 Register GSR0 – SPI status register

Table 50 **Register assignment of GSR0**

b7	b6	b5	b4	b3	b2	b1	b0
reserved					adc_result_ready	reserved	

Reset value: 0x0000

Read-only

Table 51 **Signal table of GSR0**

Signal name	Size	Function
adc_result_ready	1	**ADC result ready** This is a flag for completed conversion. It is cleared by reading out any result register (reg38–53); ADC clock must be enabled for that.

The global status register GSR0 is sent on spi_do_o at the same time as the address and the read/write bit are sent on spi_di_i, MSB leading.

3 Analog to digital converter

3.1 ADC conversion sequence

An ADC conversion consists of four different phases.

3.1.1 Enable bandgap

The bandgap is enabled by setting the bandgap_enable bit in reg34. This can be done simultaneously with clk_enable. The bandgap can be enabled or disabled independently of all other modules.

The start-up time of the bandgap is temperature and device dependent. Enabling of the ADC is not allowed before the bg_up flag (reg36) is read out as high.

3.1.2 Enable local ADC clock

The local clock generator is enabled by setting the clk_enable bit in reg34, without setting any other bits, except bandgap_enable.

3.1.3 Enable ADC

Before enabling the ADC block, the local clock and the bandgap must be available.

reg34 bit adc_enable enables the ADC. The adc_ready bit high indicates a finished start-up of the ADC. Conversion can not be started before adc_ready = "1".

3.1.4 Start ADC conversion

3.1.4.1 Single conversion

A conversion is started by SPI write command into reg35, independent from the written data. During a running conversion, no further changes of these bits are allowed.

The ADC:

- starts a sampling phase
- starts a conversion phase
- updates the corresponding result register
- sets adc_result_ready bit to "1".

3.1.4.2 Sequential conversion

A conversion sequence for all input channels can be requested by writing reg35 with chnr_all set to "1". In this case the ADC performs:

- conversion of all 16 channels consecutively and update of the corresponding result registers
- adc_result_ready bit set to "1".

The adc_result_ready bit within GSR0 is cleared by reading any of the result registers (reg38–53).

3.2 ADC configuration

3.2.1 Analog input channel gain

By setting bit lv_gain the gain for the analog input channels can be selected.

- lv_gain = 0: Full-scale analog input voltage = 1.613 V
- lv_gain =1: Full-scale analog input voltage = 1.21 V

3.2.2 Analog input voltage sampling

During the first phase, the analog input voltage is sampled onto the DAC capacitor. This is called the sampling phase. Sampling time is fixed to 16 clock cycles.

3.2.3 ADC phases

The physical resolution is 8-bit.

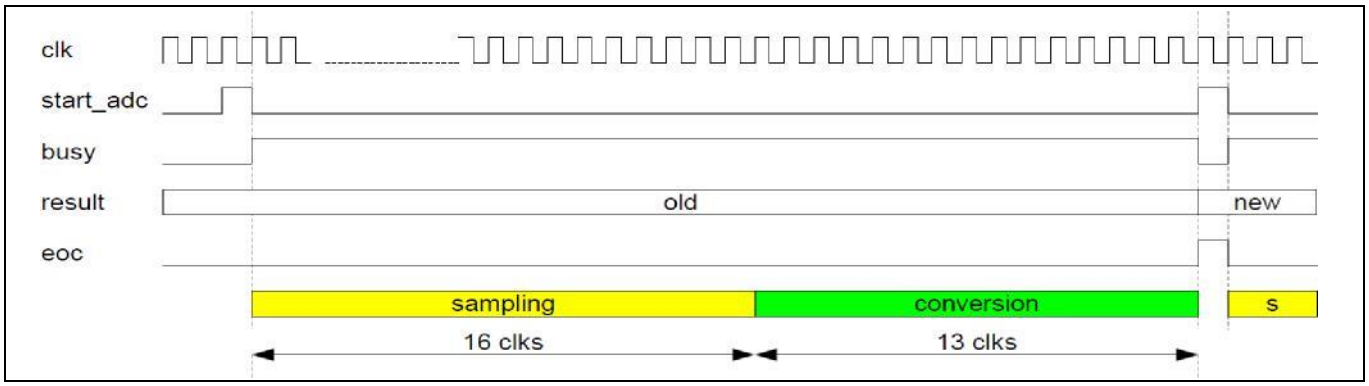

Figure 13 Timing diagram

3.3 Conversion time

An example formula for calculation is provided below. Twelve additional clock cycles are needed for post-calibration.

Sampling time is 16 clock cycles. Distribution time (=actual conversion) is 13 clock cycles.

The ADC clock is generated internally and is dependent on temperature and chip sample (min. 15 MHz, max. 50 MHz).

$t_{conv} = (t_{sample} + t_{distrib} + t_{epcal}) * t_{adc\_clk} = (16 + 13 + 12) * t_{adc\_clk}$

$t_{conv\_min} = (t_{sample} + t_{distrib} + t_{epcal}) * t_{adc\_clk\_50M} = (16 + 13 + 12) * (1/50e6) = 0.82\ \mu s$

$t_{conv\_max} = (t_{sample} + t_{distrib} + t_{epcal}) * t_{adc\_clk\_15M} = (16 + 13 + 12) * (1/15e6) = 2.73\ \mu s$

3.4 ADC power-down sequence

In case a low current consumption mode is required a full ADC power-down can be invoked in two phases:

1) Disable ADC by setting adc_enable to "0". The clock must still be running to enable the finite state machine to switch to a defined state.

2) Disable clock by setting clock_enable to "0".

Bandgap can be disabled separately by setting bandgap_enable to "0". This can be done after step 1 or after step 2.

4 Detector

4.1 Detector block diagram

The detector is connected to the single-ended I and Q signals of the ABB. It consists of Schmitt triggers and a pulse generator within the analog part, and a configurable evaluation block within the digital part. Also, the threshold voltage of the Schmitt triggers is configurable.

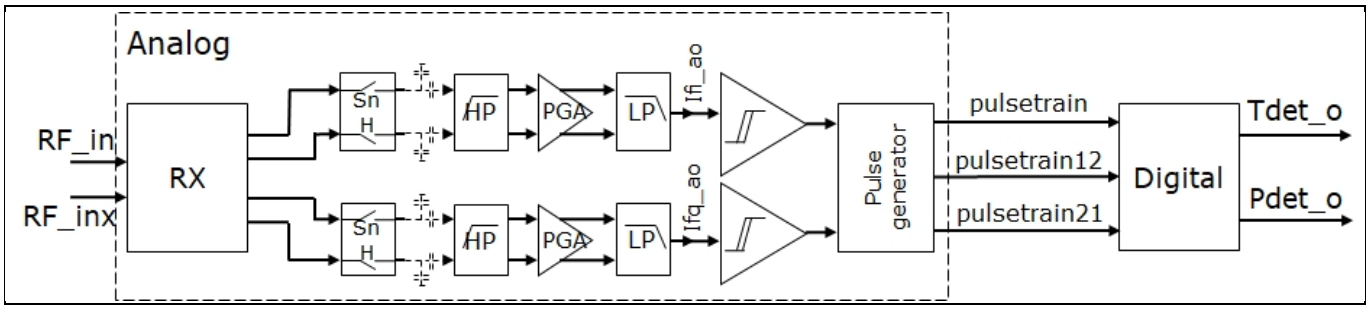

Figure 14 Detector block diagramm

4.2 Schmitt trigger

The Schmitt trigger is connected directly to the IQ outputs. Switching threshold is set by reg10[7:4]. Rough setting can also be done with QS2, which controls reg10[7:6].

4.3 Pulse generator

The pulse generator gets the rectangular signal from the Schmitt trigger. Figure 15 shows the behavior of output signals for typical input signals. V_{cmp_Ifi} has a phase shift of 90 degrees compared to V_{cmp_IfQ}.

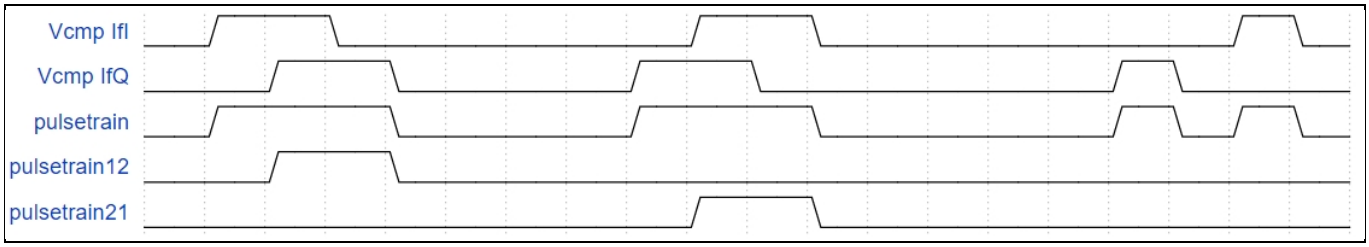

Figure 15 Pulse generator wave diagram

When Ifi becomes H before IfQ an approaching target is detected. pulsetrain and pulsetrain12 are set to H. When IfQ becomes H before Ifi a departing target is detected. pulsetrain and pulsetrain12 are set to H. Single events on Ifi or IfQ are not valid signals because I and Q have at any time the 90-degree phaseshift. These single events could occur due to disturbances or due to a high threshold voltage and a signal voltage which tops the threshold voltage only by a small amount. As the pulsetrain output is realized with an or-gate, pulsetrain becomes H for every detected H on the inputs Ifi and IfQ.

4.4 Digital evaluation

The digital block within the detector is responsible for evaluating the input signals from the pulse generator and for setting of T_{det}/P_{det} outputs of the BGT60LTR11AiP.

Target detected (T_{det}) ouput is low active; it is set to L when enough events on pulsetrain are counted. Phase detected (P_{det}) output is used to show the direction of the detected target. It is only valid in case of a detected target and it is set to H for approaching targets and L for departing targets.

There are four settings that influence detection behavior:

- Holdtime – bb_det_hold, bb_det_hold_mul: reg10[11:8]
- Detector hit counter setting – bb_det_cnt: reg[1:0]
- Detector hit counter reset setting – bb_det_cnt_reset[3:2]
- Detector direction counter hit setting – bb_det_dir_divcnt[4]

The digital detector is switched on 50 ms after setting of the bb_dig_det_en to allow settling of the BB circuit reg1[7].

4.4.1 Holdtime

The holdtime defines the length of the L-pulse of T_{det} when a target is detected. In case another target is detected during this low pulse, the holdtime starts running again. Therefore T_{det} is stable at low when holdtime is longer than the time needed for detection.

It can be configured in reg10[7:4] in a range from 10 ms to 8 min. Rough setting is possible with QS3, which controls reg10[7:6].

4.4.2 Detector hit counter setting

The hit counter setting defines the number of events (=rising edges on pulsetrain) which are needed before T_{det} is set. As soon as the number of events is reached, T_{det} is set to low and the counter is reset back to zero, waiting for the next events to count. Its range is 10 to 80 events.

4.4.3 Detector hit counter reset setting

To avoid false detection due to disturbances, there is also a timeout implemented. The timeout is a timespan defined by hit counter reset setting. Its range is 10 to 200 ms. In case there is no event in this timespan the target counter and the direction counter are reset to zero.

4.4.4 Detector direction counter hit setting

The two direction counters are used for detecting the target direction. The IfI counter counts pulsetrain12 events and the IfQ counter counts pulsetrain21 events. Generally also the hit counter setting is used for these counters. As soon as one of the two counters reaches the set limit, the output P_{det} is set accordingly. In case the IfI counter reaches the limit, it is set to H and in case the IfQ counter reaches the limit it is set to L. Furthermore, if either of the two counters reach the limit, both are reset to zero.

The direction counter hit setting bit selects if the hit counter setting is used for the limit or the half of the hit counter setting.

- Example 1: bb_det_cnt = 10b = 40 events; bb_det_dir_divcnt = 0b (=full count needed)
  so 40 events needed for target detection, 40 events in one direction needed for direction detection
- Example 2: bb_det_cnt = 01b = 20 events; bb_det_dir_divcnt = 1b (=half count needed)
  so 20 events needed for target detection, 10 events in one direction needed for direction detection

4.4.5 Example wave diagram with outputs and counters

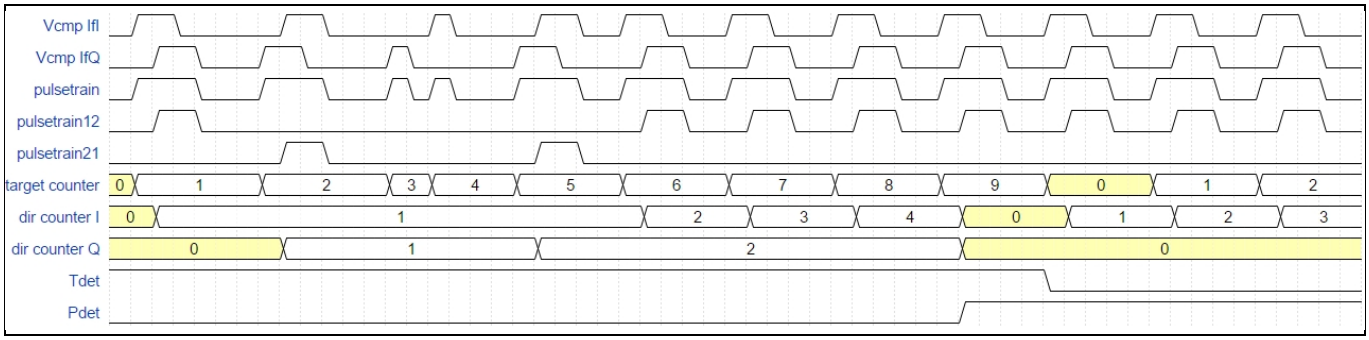

Figure 16 Detector wave diagram example

Figure 16 shows an example of a detection, including the internal counters. The hit counter is set to 10 and the direction divider count to 1, meaning that five events are needed for direction detection. The starting condition for the outputs is no detection (T_{det} = H) and P_{det} = L, which means the last target was a departing one.

The incoming waveform on V_{cmp} IfI and V_{cmp} IfQ is the waveform of an approaching target. The target counter counts each event on pulsetrain. With the tenth event, the counter is reset to zero and the target detection is indicated on the output T_{det} by setting it to L. The direction counters are not influenced.

With the fifth event on pulsetrain12 the direction of the target is detected. Both direction counters are set to 0 and P_{det} is set to H, that means the target is approaching. The target counter is not influenced.

Revision history

Document version	Date of release	Description of changes
V1.0	2020-09-09	First version
V1.1	2020-10-06	Added autonomous mode

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-10-05

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AN625

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.