BGT60LTR11AIP MMIC User Guide

60 GHz Radar

About this document

Scope and purpose

This application note is intended to provide more details on how to use BGT60LTR11AIP in an actual application as compared to the datasheet.
The datasheet gives technical data and limits of the device itself but it is not explaining how to operate the device in greater detail. This application note takes care of this issue.
The reader will find here:

- Discussion of all different building blocks
- How to operate the different blocks
- Settings of the SPI registers are grouped on topic including truth tables
- Additional measurement data showing behavior over temperature

Intended audience

Hardware engineers and software engineers working on designs with Infineon's BGT60LTR11AIP.

Table of contents

1 Introduction to BGT60LTR11AIP

The BGT60LTR11AiP is a fully integrated microwave motion sensor including antenna structures, configurable built-in detectors and a state machine allowing for fully autonomous operation. It is designed to operate as a Doppler motion sensor between 61 to 61.5 GHz frequency band.

An integrated frequency divider with a phase locked loop (PLL) provides voltage-controlled oscillator (VCO) frequency stabilization and allows for continuous wave (CW) operation. The device supports three operation modes, fully autonomous, semi-autonomous, and SPI mode. The different modes can be selected via hardware preset pins.

The BGT60LTR11AIP has an integrated low phase noise push-push VCO for the high frequency signal generation. The transmit section consists of a medium power amplifier with configurable/adjustable output power, which can be controlled via serial peripheral interface (SPI). The transmitted power is monitored by integrated power detectors. The packaged monolithic microwave integrated circuit (MMIC) features integrated broadbeam antennas for maximum area coverage.

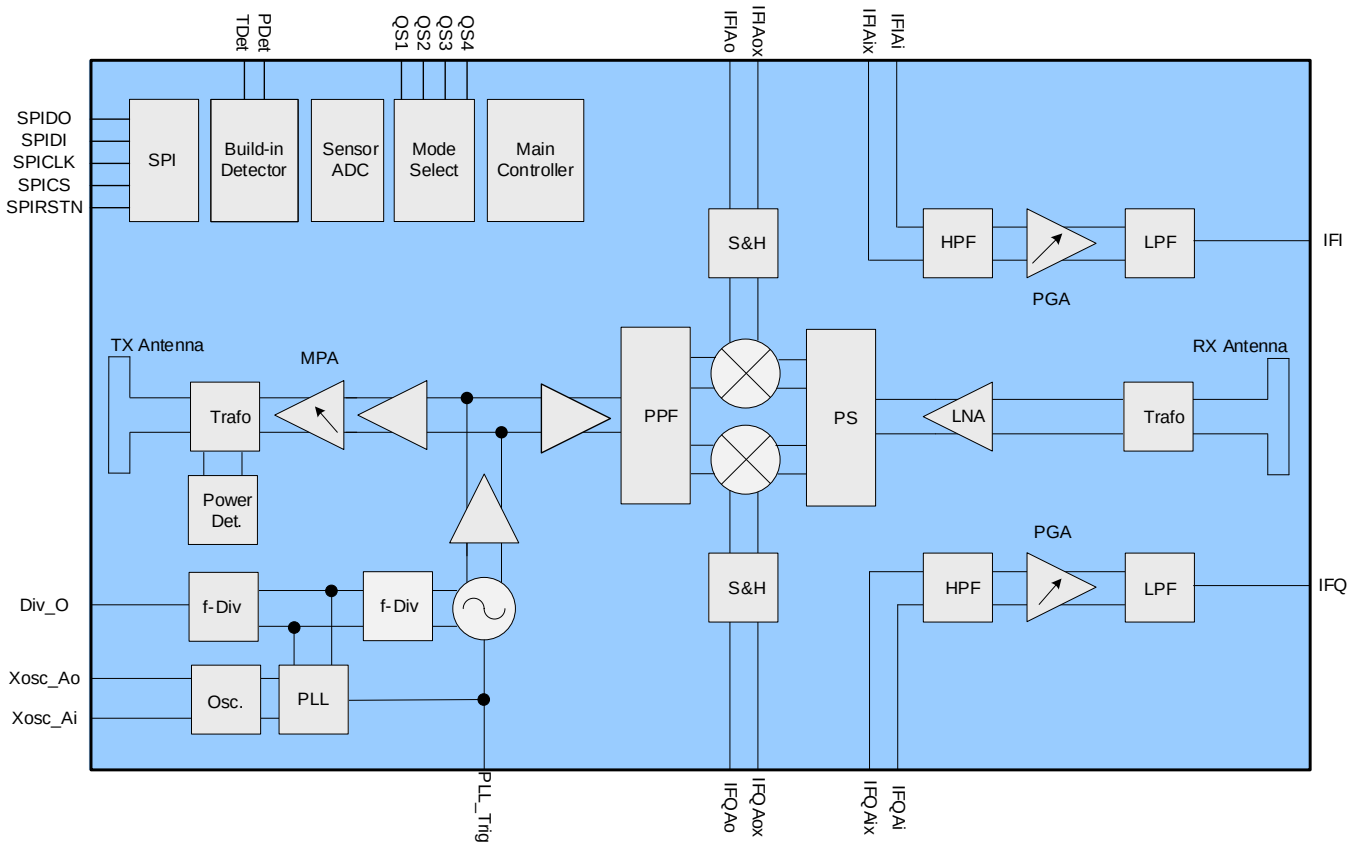

Figure 1 BGT60LTR11AIP block diagram

2 Main Controller

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

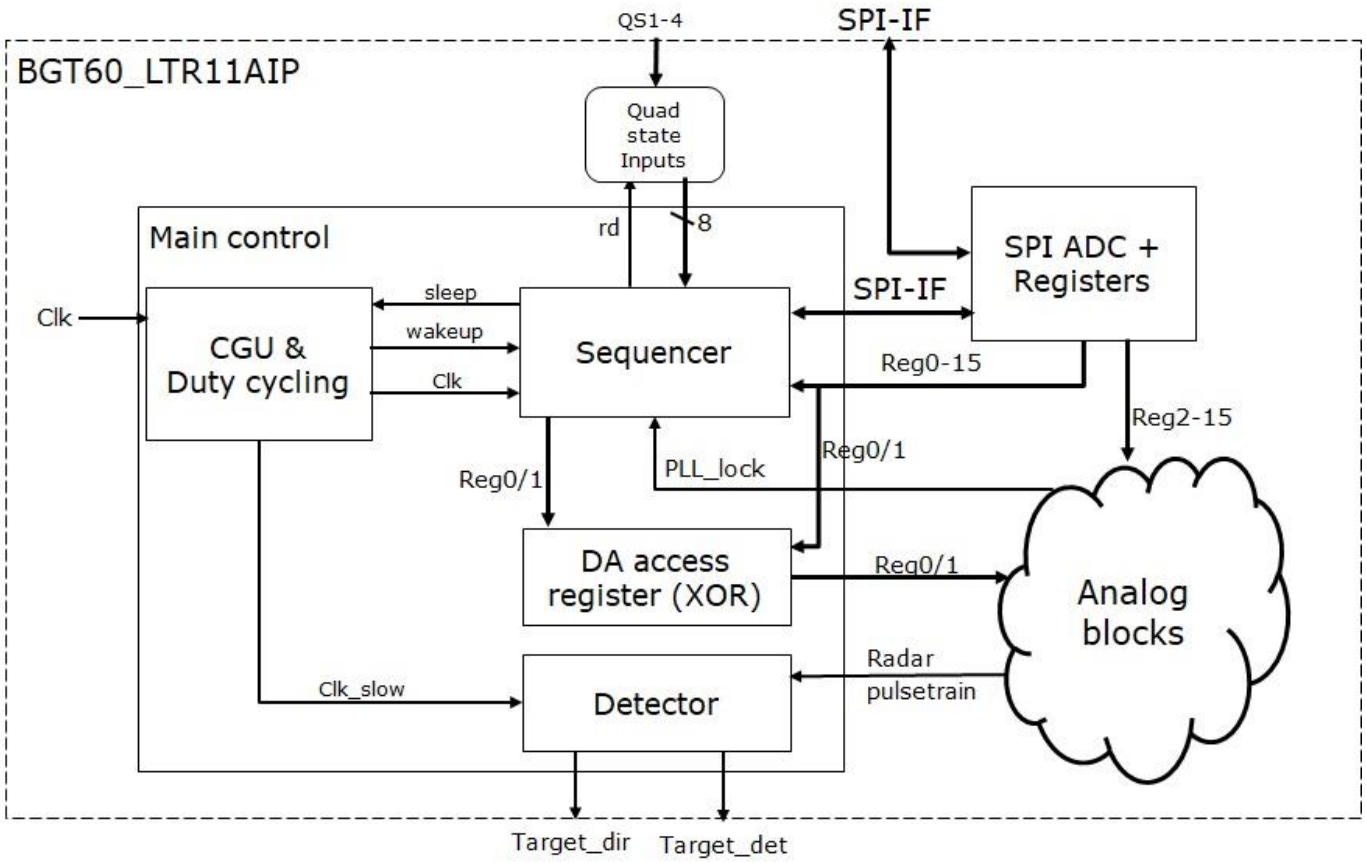

Figure 2 Main controller block diagram

The interpretation of the (inverting) output pads voltage can be configured by fuse[0] to support any of these modes:

- Default direction information set to “departing”
- Default direction information set to “approaching”

The truth table is shown here:

		Output pad voltage			
		Mode 1 (fuse[0] = 0)		Mode 2 (fuse[0] = 1)	
Motion	Appr. / Dep.	TDet	PDet	TDet	PDet
No	Dep	high	low	high	high
No	Appr	high	low	high	high
Yes	Dep	low	low	low	low
Yes	Appr	low	high	low	high

The BGT60LTR11AIP provides four quad states 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.

2.1 “Advance mode” and Quad state inputs

On reset of the digital main_controller and during the init-sequence some chip input pins are sampled to determine the the chip should start with.

2.1.1 “Advance mode”

When this pin is 0 when the digital main_controller wakes up from reset, the chip boots in Basic mode.

When the pin is kept 1 during chip boot and QS1 is either GND or OPEN, pins SPI_CLK and SPI_MOSI are also sampled to determine the PRT (dc_rep_rate). In addition pins QS2 and QS3 are evaluated by the ADC and converted in 4-bit values each after each “mean window”.

Table 1 **PRT in Advance-mode**

PLL_TRIG	SPI_MOSI	SPI_CLK	dc_rep_rate
0	*	*	1 / 2 (fuse dependent)
1	0	0	1
1	0	1	3
1	1	0	0
1	1	1	2

2.1.2 Quad state basics

The quad state inputs allow to configure four different states with one input pin. Table 2 show 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 2 **States of a quad state input**

Pad	b1	b0
ground	0	0
open	0	1
100kΩ to V_{DD}	1	0
V_{DD}	1	1

2.1.3 QS 1

QS 1 is used to select the mode of the chip.

Table 3 **QS 1**

Pad	b1	b0	Operating mode
ground	0	0	Continuous wave (CW) mode
open	0	1	Autonomous pulsed mode

100kΩ to V_{DD}	1	0	SPI mode with external 9.6MHz clock enabled
V_{DD}	1	1	SPI mode

2.1.4 QS 2

QS 2 is used to select the detector threshold value, with the same radar cross section of a target, a lower detector threshold value correspond to a higher detection range, it is written into register 2 described in Table 34.

Table 4 **QS 2**

Mode	Pad	b1	b0	Detector threshold
Basic mode	ground	0	0	80
Basic mode	open	0	1	192
Basic mode	100kΩ to V_{DD}	1	0	480
Basic mode	V_{DD}	1	1	2560
Advance mode	1*VDD/16 – 2*VDD/16	*	*	66
Advance mode	2*VDD/16 – 3*VDD/16	*	*	80
Advance mode	3*VDD/16 – 4*VDD/16	*	*	90
Advance mode	4*VDD/16 – 5*VDD/16	*	*	112
Advance mode	5*VDD/16 – 6*VDD/16	*	*	136
Advance mode	6*VDD/16 – 7*VDD/16	*	*	192
Advance mode	7*VDD/16 – 8*VDD/16	*	*	248
Advance mode	8*VDD/16 – 9*VDD/16	*	*	320
Advance mode	9*VDD/16 – 10*VDD/16	*	*	384
Advance mode	10*VDD/16 – 11*VDD/16	*	*	480
Advance mode	11*VDD/16 – 12*VDD/16	*	*	640
Advance mode	12*VDD/16 – 13*VDD/16	*	*	896
Advance mode	13*VDD/16 – 14*VDD/16	*	*	1344
Advance mode	14*VDD/16 – 15*VDD/16	*	*	1920
Advance mode	15*VDD/16 – 16*VDD/16	*	*	2560

2.1.5 QS 3

QS 3 is used to select the holdtime of the `Tdet_o` output, this defines how long the output status will be kept after a target is detected, it is written into register 10 decribed in Table 34.

Table 5 **QS 3**

Mode	Pad	b1	b0	target_det holdtime
Basic mode	ground	0	0	16/32/64/128 ms (depends on dc_rep_rate)
Basic mode	open	0	1	1 s
Basic mode	100kΩ to V_{DD}	1	0	10 s
Basic mode	V_{DD}	1	1	1 min
Advance mode	0*VDD/16 – 1*VDD/16	*	*	100 ms
Advance mode	1*VDD/16 – 2*VDD/16	*	*	500 ms
Advance mode	2*VDD/16 – 3*VDD/16	*	*	1 s

Advance mode	3*VDD/16 – 4*VDD/16	*	*	2 s
Advance mode	4*VDD/16 – 5*VDD/16	*	*	3 s
Advance mode	5*VDD/16 – 6*VDD/16	*	*	5 s
Advance mode	6*VDD/16 – 7*VDD/16	*	*	10 s
Advance mode	7*VDD/16 – 8*VDD/16	*	*	30 s
Advance mode	8*VDD/16 – 9*VDD/16	*	*	45 s
Advance mode	9*VDD/16 – 10*VDD/16	*	*	1 min
Advance mode	10*VDD/16 – 11*VDD/16	*	*	90 s
Advance mode	11*VDD/16 – 12*VDD/16	*	*	2 min
Advance mode	12*VDD/16 – 13*VDD/16	*	*	5 min
Advance mode	13*VDD/16 – 14*VDD/16	*	*	10 min
Advance mode	14*VDD/16 – 15*VDD/16	*	*	15 min
Advance mode	15*VDD/16 – 16*VDD/16	*	*	30 min

2.1.6 QS 4

QS 4 is used to select the device operating frequency by configuring the PLL. Frequency is also dependent on the Japan e-fuse.

Table 6 **QS 4**

Pad	b1	b0	Japan e-fuse	VCO frequency
ground	0	0	1	61.1 GHz
open	0	1	1	61.2 GHz
100kΩ 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
100kΩ to V_{DD}	1	0	0	60.8 GHz
V_{DD}	1	1	0	60.9 GHz

2.2 Power up and sequencing

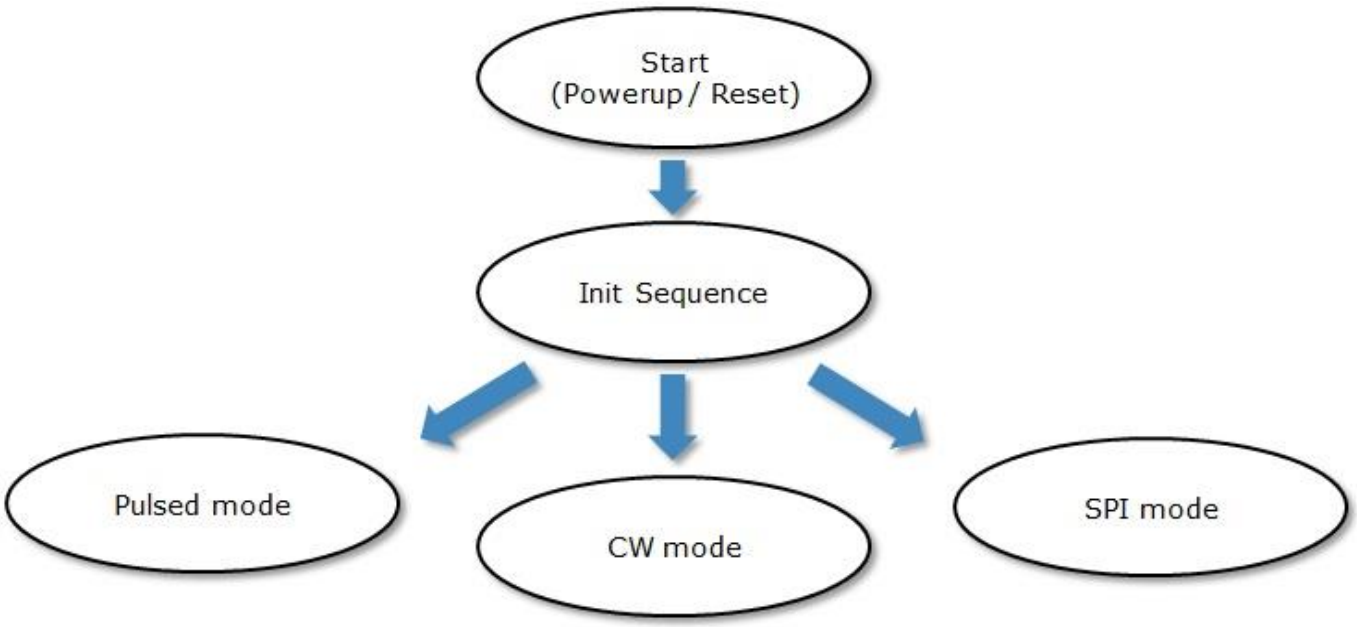

Figure 3 State diagram overview

Depending on setting of QS1 one of three available modes is selected. At startup the internal main controller has control over the SPI interface. So it is not recommended to program the chip from external during that phase or while in an automatic (pulsed- or CW-) mode. So SPI activity is recommended only while in SPI-mode.

This is because 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 interface. So the pad must be set to 1 if SPI interface is not used or an external controller is not existing.

2.2.1 Power up

Figure 4 shows internal signals relevant for power up.

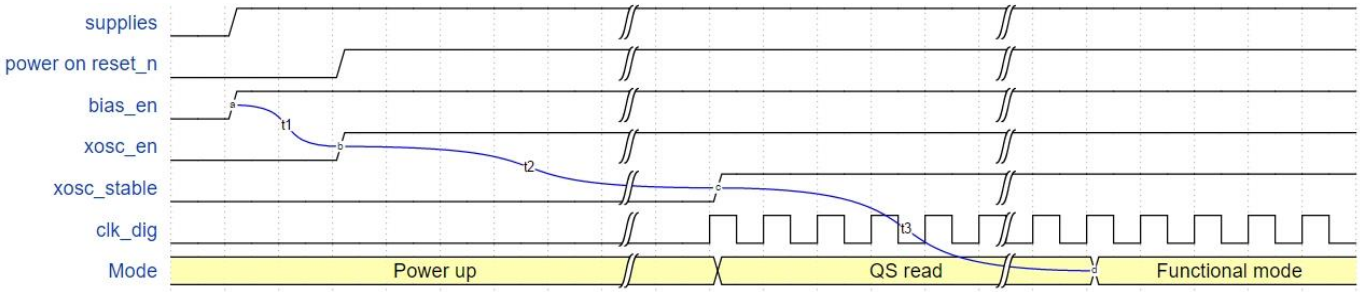

Figure 4 Power up

Supply ramp needs to be shorter than 400 us. Bias_en is connected directly to the supply therefore it ramps simultaneously. The integrated power on reset makes sure that 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 us. This is fulfilled when time for ramping supply is as defined.

The oscillator needs the time t_2 to get stable and activate clock for 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 SPI commands from outside in SPI mode, in pulsed mode or CW mode it takes 25 µs more.

2.2.2 Init sequence

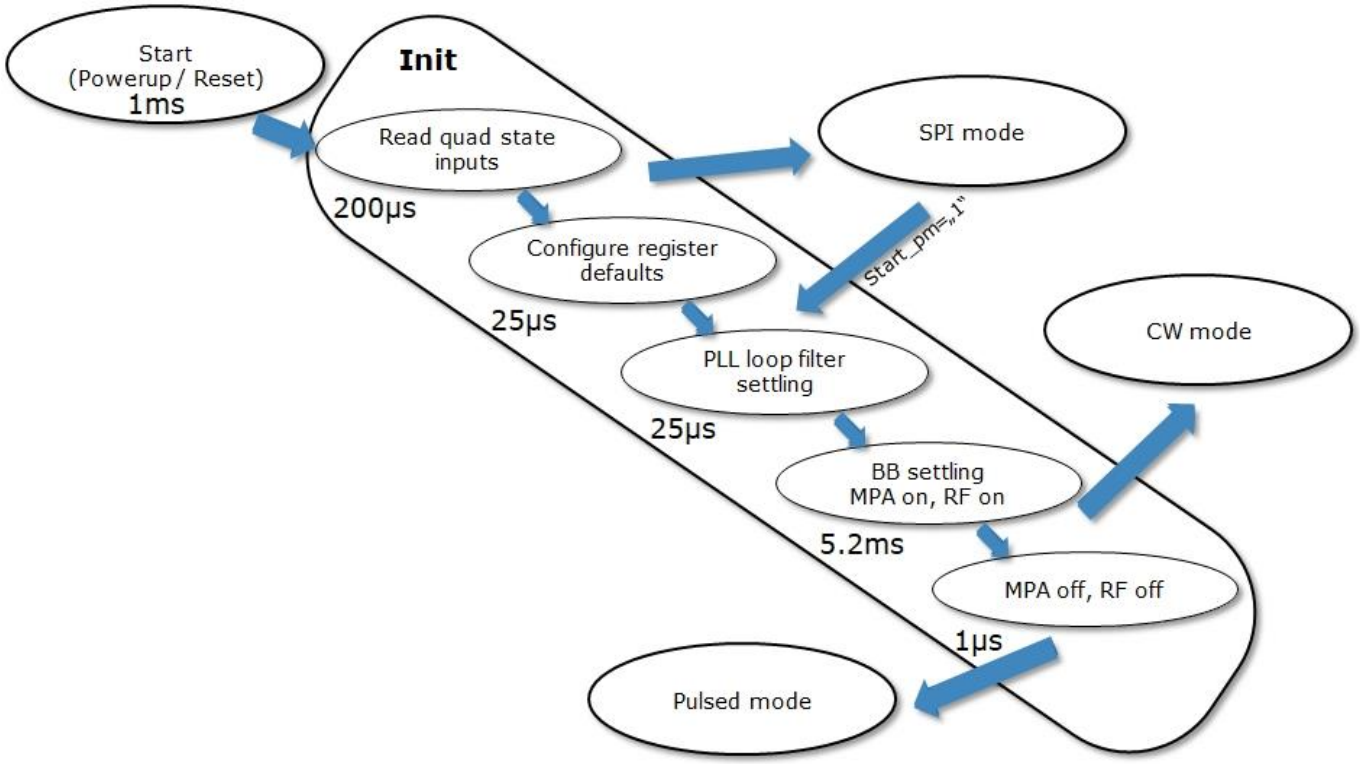

Figure 5 Init sequence

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

1. Read quad state inputs
   Quad state inputs needs 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 interface is handed over to the external microcontroller. The main controller switches into SPI mode.
2. For pulsed and CW mode now the default values for the configuration registers are prepared and written using the SPI interface of the SPI ADC block internally.
3. PLL is started, medium power amplifier MPA is not activated during the loop filter settlement. After 20us also the MPA is activated, RF is also running. The next 5ms are used for 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 is switched off and the main controller switches into pulsed mode.

The status bit init_done (reg56[13]) is set when leaving init sequence.

Table 7 **Init sequence in detail**

Nr	Command	Description
1	write reg1 0x0100	set bit qs_rd_en

Nr	Command	Description
2	2*read reg55	Read fuse values, 2 times to provide enough clocks to read out fuses within SPI ADC before reading the register
3	wait 200 µs	Wait 200 µs before reading quad state inputs
4	read quad state inputs	Read & end init sequence if mode = SPI mode
5	write reg1 0x0000	Reset bit qs_rd_en
6	spiwrite reg4 reg4_init	Write defined default value (partly calculated from efuses)
7	spiwrite reg5 reg5_init	Write defined default value (partly calculated from efuses)
8	spiwrite reg6 reg6_init	Write defined default value
9	spiwrite reg7 reg7_init	Write defined default value (partly calculated from pin pll_trigger_i, efuses, SPI_CLK and SPI_MOSI)
10	spiwrite reg8 reg8_init	Write defined default value (partly calculated from efuses)
11	spiwrite reg9 reg9_init	Write defined default value (partly calculated from efuses)
12	spiwrite reg2 reg2_init	Write defined default value (calculated from pin pll_trigger_i, efuses and quad states – analog read-in in Advance mode)
13	spiwrite reg10 reg10_init	Write defined default value (calculated from pin pll_trigger_i, and quad states – analog read-in in Advance mode)
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
20	write reg0 0x373F	Set mpa_en
21	wait MPA 2 sample enable	Wait defined time mpa enable to sample&hold – mpa2sh_dly (reg7[5:4])
20	write reg1 0x1037	Set bb_sample_en
21	wait 5 ms	Wait time for baseband 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 100ns
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

Nr	Command	Description
		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]). Afterwards the pulsed sequence is started.

2.2.3 Pulsed sequence

In pulsed mode the device is active only a short time followed by a time where VCO, RF and PLL is off. Baseband keeps 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 500µs and an on time of 14µs which consists of about 9µs needed for startup of RF and PLL + 5µs default sample time.

Table 8 **Pulsed sequence in detail**

	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 S&H	Wait defined time mpa enable to sample&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 100ns
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

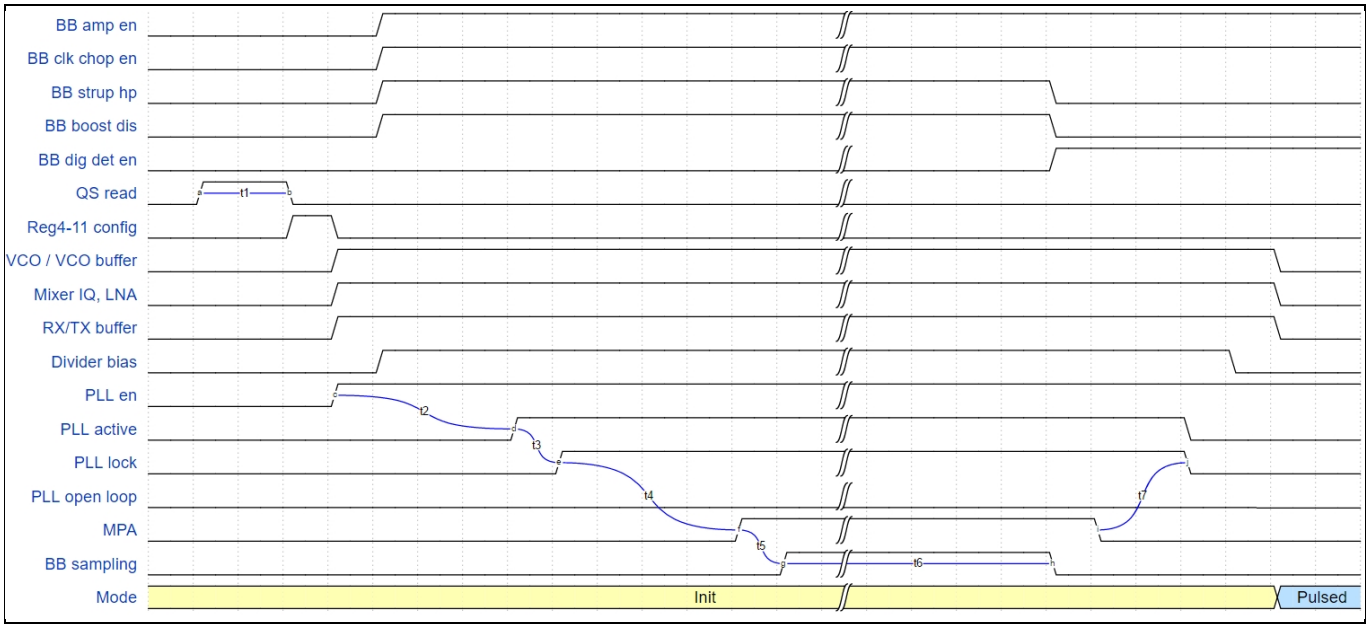

Figure 6 Pulsed mode – init phase

Figure 6 shows the wavediagram of the init phase in pulsed mode for all control signals.

t_1: 200 µs, time quad state read input of quad states needs to be one before reading them

Before switching on any block the efuses 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 baseband settling

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

Now digital detector is switched on, it takes another 50ms 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&Hold / MPA / PLL / Divider bias / RF, each step has to take 100ns, after MPA t_7 is needed. The signal pll_clk_gate_en has to be the same as pll_active.

Registers can be programmed during active pulsed mode from external, but 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 – reg6) must not be changed when pll_en=1 (reg0[8]) or pll_clk_gate_en=1 (reg0[10])
- Digital detector settings (reg2, reg10) 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 powerup phase is skipped. The next wakeup will happen with the next wakeup pulse.

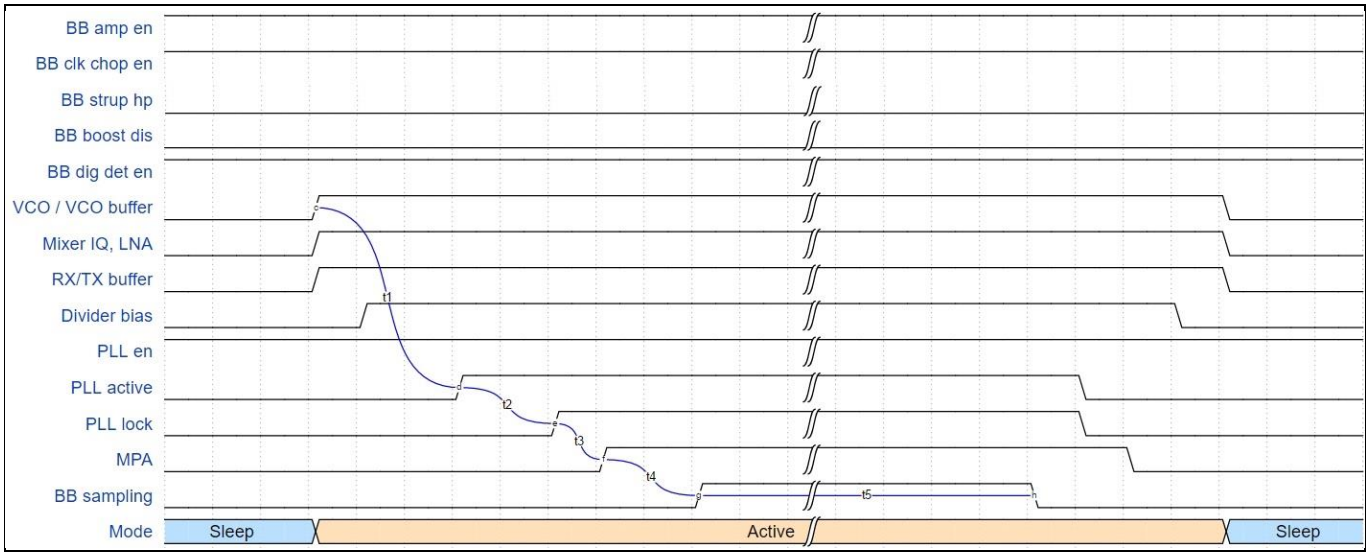

Figure 7 Pulsed mode – Active phase

Figure 7 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: 300-400 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&Hold / PLL / MPA / Divider bias / RF, each step has to take 100ns. The signal pll_clk_gate_en has to be the same as pll_active.

As soon as the pulse is finished the ADC is started and the IF signals are converted and evaluated by the digital detector.

2.2.4 CW sequence

In continuous wave mode the device is active as configured with e-fuses and quad state inputs. The Main controller only uses the ADC and evaluates the IF signals in the digital detector. 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 9 **CW sequence in detail**

	Command	Description
1	write reg1 0x10B3	set bb_dig_det_en, reset bb_strup_hp
2	end	

Figure 8 shows CW mode including init phase.

Init phase until end settling time for baseband is the same as for pulsed mode. As in CW mode baseband and sampling is on continuously bb_strup_hp is set to 0. As in CW mode baseband and sampling is on continuously bb_strup_hp is set to 0.

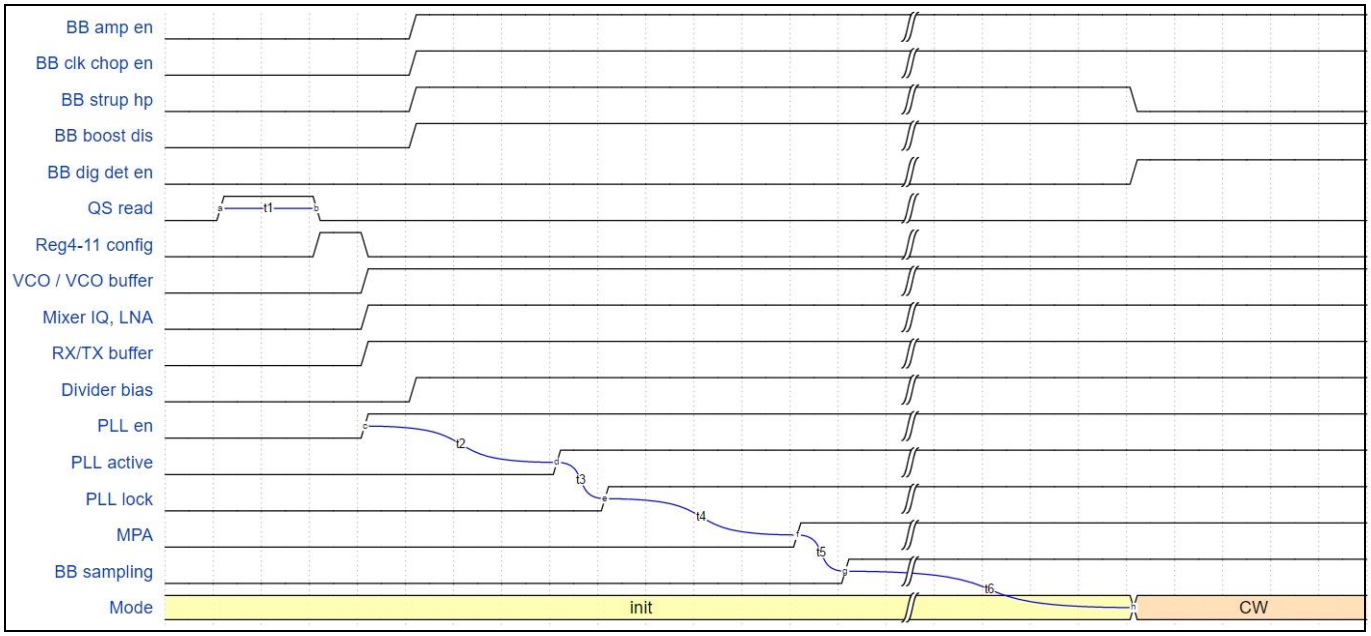

Figure 8 CW mode

2.2.5 SPI mode sequence

This is the manual mode, main controller is inactive and reg0/1 are set to all off by main controller. The SPI sequence just has to switch off quad state inputs. Depending on quad state input 1 external 9.6MHz clock is enabled (qs1=”10”) or not (qs1=”11”).

Table 10 **SPI mode sequence in detail**

	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 or CW-mode afterwards. This can be done by setting start_pm (reg15[14]) to 1. This enters pulsed-mode if reg15[12] is 0, CW-mode otherwise).
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, 16-32 further clocks are delivered to allow the external component to go to a sleep state.

2.2.6 Detector

When RF blocks are switched on, detector can be activated. It takes another 50ms after enabling of detector for settling of baseband. During this time the outputs Tdet and Pdet are kept in inactive state.

2.3 Overview dynamic control signals

Table 11 **Overview dynamic control signals**

	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	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 (=register 0/1). The main controller is able to do sequencing with full clock speed so it takes 100ns for each register access. For all other registers the main controller has to use the SPI interface which takes about 25 clock cycles (=2.5 µs).

Figure 13 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 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 main controller continuously. Programming it with “1” would just invert the bit.

3 SPI interface

SPI – Serial Peripheral Interface

- 7bit continues address space
- Fixed payload of 16bits
- Chip-Select (Slave Select) active in low state
  Has to be “1” unless an SPI access is done by external SPI master. Such an access is recommended only in SPI-mode. Otherwise there may be conflicts on the SPI interface as this is also used internally by the main_controller.
- With SPI_CS=1 the data output SPI_DO (aka MISO) is High-Z.

In order to avoid issues cause by the conflicts mentioned above, here are some recommendations:

- When the SPI is not used, keep `spi_cs_n_i` high. Otherwise the internal sequencer cannot continue and operation stops.
- While chip is working autonomously (pulsed / CW) the following timing constraints must be considered. **In order to make spi_do_o available it’s highly recommended to set bit “miso_drv” (reg15[6])!**
  - During accesses `spi_cs_n_i` must not be driven low too long (1.6us TBD). That would prevent the internal use of SPI and break the timing in the sequencer. This is mostly relevant with low data rates (below 16Mbps) or with burst accesses.
  - When starting an access the following arbitration protocol must be used:
    1. Initially `spi_cs_n_i` is driven high. Minimum for >=100ns.
    2. Wait until `spi_do_o (MISO)` is low (high means the internal sequencer is using the SPI).
       That should not take long. However if something is completely broken a timeout may help in detecting such issues.
       Skip this step if “miso_drv”=0!
    3. Drive `spi_cs_n_i` low (try to reserve the bus).
    4. Wait for >=100ns (the time needed for synchronization of `spi_cs_n_i`).
    5. Check `spi_do_o (MISO)` again. If it’s high now (sequencer has just started SPI also), goto step 1.
       Otherwise continue to step 6.
    6. Do the actual data transmission. Don’t spend too much time here (see above)!
    7. Drive `spi_cs_n_i` high (release the bus).
- As an alternative to the restrictive timing and arbitration scheme during active mode above it’s also possible to use the SPI during the slots when it’s not used internally. That means watching the periodic pattern (period is the sample rate) on `spi_do_o` and align external accesses accordingly.

3.1 SPI Interface 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. This timing on spi_do_o can be changed by setting reg15[8] to 1. In that case spi_do_o is delayed by half a SPI clock cycle (therefore changes on the falling edge instead). This leaves more hold-time (but less setup-time) for the external SPI master.

3.1.1 SPI write mode

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

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). 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.

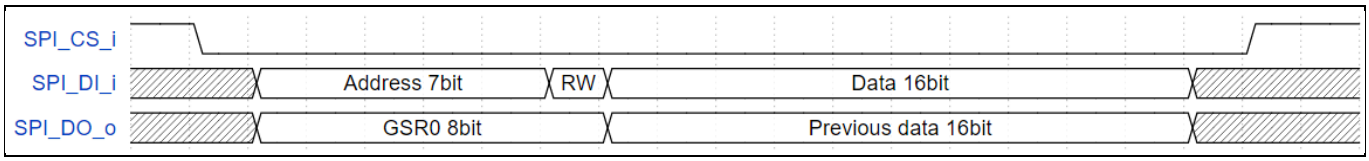

Figure 9 SPI write – MSB first anytime, RW='1'

3.1.2 SPI read mode

A read access starts after the falling edge of spi_cs_n_i with transfer of the 7bit address, MSB first. The followed eight' bit (RW = read/write bit) is "0" indicating a read access. The following 16bit 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.

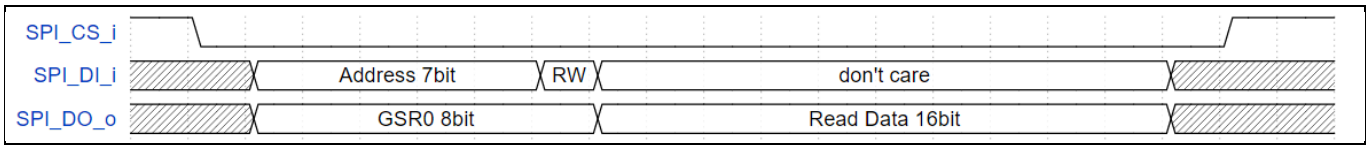

Figure 10 SPI read – MSB first anytime, RW='0'.

3.1.3 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 bit fields and is shown in Table 12.

Burst command examples:

Burst command for start read from register 4: 0xFF08

Burst command for start write from register 7: 0xFF0F

Table 12 **Burst mode command**

Bit field	Bit Width	Bitfield Name	Description
15:9	7	addr	Address to start the burst
	1	bit8	bit to fill the 16bit command = 1
7:1	7	saddr	Burst mode starting address
0	1	rwb	Burst mode read/write 0 - burst read 1 - burst write

3.1.3.1 SPI burst access

After the start condition the 16bit burst mode command is sent from the SPI master on `spi_di_i`. At the same time, the status register GSR0 (8bit) and 8bit 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 read sequence, the SPI master reads from the device.

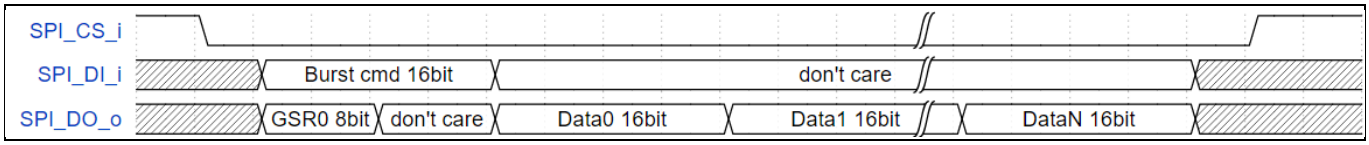

Figure 11 SPI burst read

Burst mode write sequence:

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

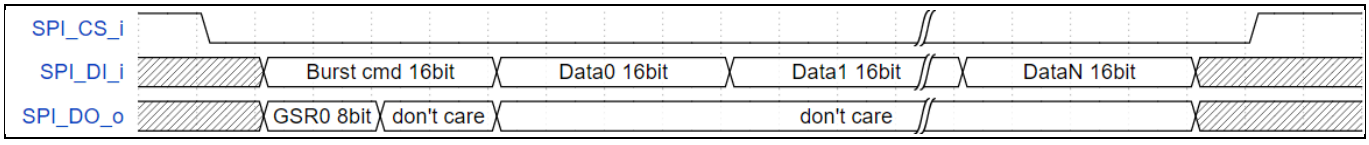

Figure 12 SPI burst write

3.2 SPI register

3.2.1 Register Overview

Table 13 **Register Overview including reset values for e-fuse=0x0000**

Register	Mode	Contents	Reset value	Value after init sequence (pulsed sleep / CW)
Reg0	RW	Control bits	0x0000	0x0900* / 0x373F*

Register	Mode	Contents	Reset value	Value after init sequence (pulsed sleep / CW)
Reg1	RW	Control bits	0x0000	0x0092* / 0x10B3*
Reg2	RW	Threshold	0x0000	dep. on QS2
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	Baseband	0x0000	0x0066 / 0x0076
Reg10	RW	Holdtime	0x0000	dep. on QS3
Reg11	RW	Reserved	0x0000	0x0000
Reg12	RW	BITE, AMUX	0x0000	0x0000
Reg13	RW	Algo 1	0x0000	0x0000
Reg14	RW	Algo 2	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 – 15	n/a	n/a
Reg55	RO	e-fuse	0x0000	0x0000
Reg56	RO	Status and chip version	0x0000	0x2000, bit(2:0) dep. on chipversion bit(15:14) dep. on QS 1
GSR0	RO	8bit 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 the init-sequence has finished. 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.

3.2.2 Direct access register

Reg0 and Reg1 are direct access registers shown in Figure 13. These bits can be controlled directly by the main controller. All other registers needs to be programmed by SPI from external optional micro controller or internally from the main controller at powerup.
As an external read access from SPI anytime 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 SPI ADC.

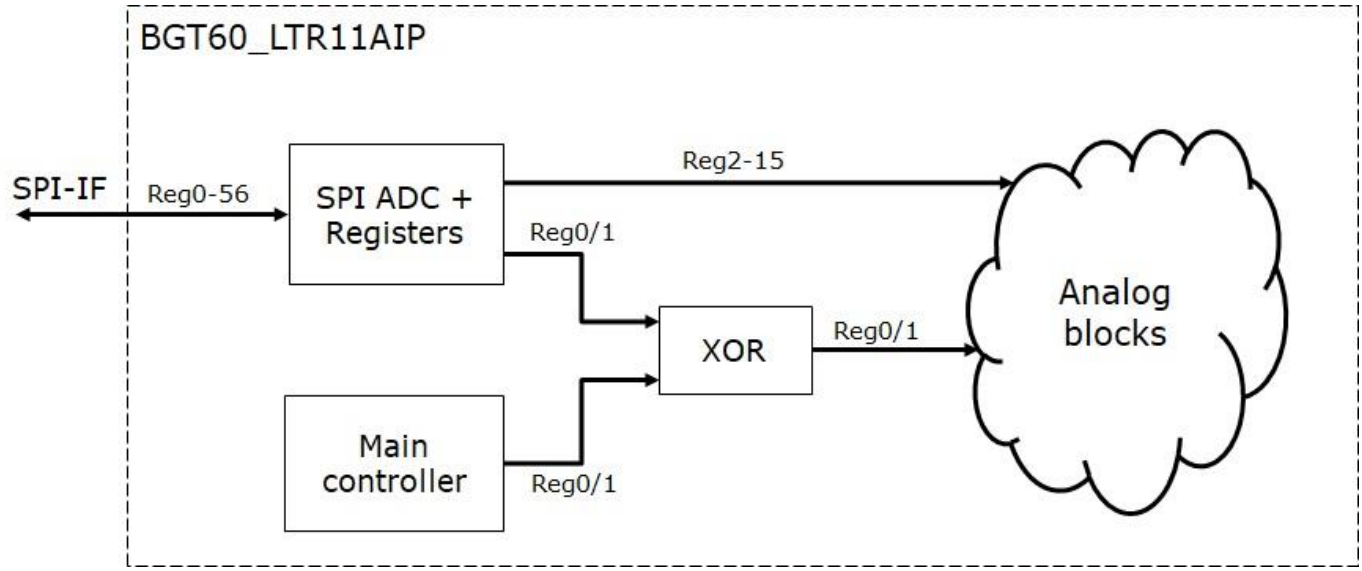

Figure 13 Direct access registers

Table 14 **XOR**

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

3.2.3 Register map bitfields

Table 15 **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]	hprt	aprt	dir_mode	thrs				
	[7:0]	thrs							
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	6	5	4	3	2	1	0
	[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]	hold							
	[7:0]								
Reg11	[15:8]								
	[7:0]								
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]	phase_win_len			mean_win_len			prt_mult	
Reg14	[15:8]	thrs_offset							
	[7:0]	dir_hyst_dis	dir_keep	hold_x32	swap_iq	autoblind		phase_thrs	
Reg15	[15:8]	soft_reset	start_pm	clk_ext_dis	start_cw	fast_phase	dir_c2_1		fastmode
	[7:0]	adc_mon	miso_drv	mot_pol	dir_pol	stat_mux			
Reg34	[15:8]	reserved							
	[7:0]	reserved					adc_en	bandgap_ en	adc_clk_en
Reg35	[15:8]	reserved							
	[7:0]	lv_gain	reserved		channel_all	channel_nr			
Reg36	[15:8]	reserved							
	[7:0]	reserved						adc_ready	bg_up
Reg38 - Reg53	[15:0]	ADC result register channel 0 - 15							

		7	6	5	4	3	2	1	0
Reg55	[15:8]	fuse_out							
	[7:0]								
Reg56	[15:8]	quad_state1		init_done					
	[7:0]			pll_ ramping	pll_pa_ active	pll_lock	Chip_version		

3.2.4 Register reg0 – Direct access register

Table 16 **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 17 **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
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

Signal Name	Size	Function
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

3.2.5 Register reg1 – Direct access register

Table 18 **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

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 baseband detector** Enables digital part of detector. After first switching on of this bit after startup/chip reset it takes 50ms 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	**Baseband sample&hold switch boost setting** 0_D: S&H gate voltage boost is enabled (pulsed mode) 1_D: S&H gate voltage boost is disabled (CW mode)
bb_clk_chop_en	1	**Enable clock chop** This bit enables continues clock signal for chopping. 0_D: Clock off 1_D: Clock on
bb_strup_hp	1	**Baseband startup boost mode** 0_D: Startup boost mode disabled 1_D: Startup boost mode enabled
bb_amp_en	1	**Enable baseband amplifier** 0_D: baseband amplifier disabled 1_D: baseband amplifier enabled
bb_sample_en	1	**Enable baseband sampling** Controls connection of mixer output to sample&hold capacitance. 0_D: Disconnected, hold phase 1_D: Connected, sampling phase

3.2.6 Register reg2 – Threshold

Table 19 **Register assignment of reg2**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
hprt	aprt	dir_mode	thrs												

Reset value: 0x0000
Value after init sequence: Depends on fuses and QS2

Table 20 **Signal table of reg2**

Signal Name	Size	Function
hprt	1	**High pulse repetition time** After init-sequence this is the inverse of the level on SPI_CS_N at boot time. 0_D: No change 1_D: PRT is constantly multiplied by prt_mult in reg13. This is for saving power, but may cause problems with direction detection.
aprt	1	**Adaptive pulse repetition time** After init-sequence this is a copy of fuse[1]. 0_D: No change 1_D: PRT is multiplied by prt_mult in reg13 as long as no target is detected. This is for saving power.
dir_mode	1	**Direction detection mode** After init-sequence this is a copy of fuse[0]. 0_D: Mode 1 Pdet=0 when Tdet=1 when no target is detected. 1_D: Mode 2 Pdet=Tdet=1 when no target is detected.
thrs	13	**Detector threshold level** Default after init sequence: Depends on QS2 This is internally divided by 32 and then corresponds to LSB of the ADC results of the IF signals.

3.2.7 Register reg4 – PLL config 1

Table 21 **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): 0x0F3A

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

Table 22 **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 bit field 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 statemachine. 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
pll_bias_dis	1	**PLL bias disable** Default after init sequence: 0_D Disables bandgap in PLL and V2I 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	**Loopfilter isolation mode** Default after init sequence: 1_D 0_D: Isolation with charge-keeping buffer enabled 1_D: Isolation with switches only

Signal Name	Size	Function
pll_lf_r4_sel	1	**Loopfilter R4 setting** Default after init sequence: 1_D / 0_D Efuse used 0_D: 12.4k 1_D: 0.1k
pll_cl_loop_pmode	1	**Closed loop in pulsed mode** Default after init sequence: 1_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 closed loop is also used for pulsed mode. 0_D: open loop mode used in pulsed mode 1_D: closed loop mode used in pulsed mode (pll_open_loop forced to 0)
pll_lf_r2_sel	1	**Loopfilter R2 setting** Default after init sequence: 1_D / 0_D Efuse used 0_D: 21.6k 1_D: 18.7k
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: 60GHz-mode cntA=21dec for 38.4MHz 1_D: 60GHz-mode cntB=20dec for 40MHz
pll_cp_icp_sel	3	**Charge pump current setting** Default after init sequence: 55µA / 30µA (7_D / 2_D) Efuse used 0_D: 20µA 1_D: 25µA 2_D: 30µA 3_D: 35µA 4_D: 40µA 5_D: 45µA 6_D: 50µA 7_D: 55µA
pll_cp_mode	1	**Charge pump bias mode** Default after init sequence: 0_D / 1_D Efuse used 0_D: Bias regulation loop active 1_D: Fix bias mode = bias regulation loop off

Signal Name	Size	Function
pll_pfd_rdt_sel	2	**PFD reset delay time select** Default after init sequence: 2_D 0_D: 175ps 1_D: 275ps 2_D: 375ps 3_D: 470ps

3.2.8 Register reg5 – PLL config 2

Table 23 **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 efuse

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

Table 24 **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 efuse FCW for fstart (4bit integer + 8bit fractional) → 2.4MHz raster @ 60GHz 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

3.2.9 Register reg6 – PLL config 3

Table 25 **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 26 **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 Efuse 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: 4.6ns
pll_ld_len	1	**Lock detection - lock assertion condition +** **Lock detection - lock delay** Default after init sequence: 0_D / 1_D Efuse 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

Signal Name	Size	Function
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

3.2.10 Register reg7 – Duty cycling, timing, pd, MPA

Table 27 **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 28 **Signal table of reg7**

Signal Name	Size	Function
dc_rep_rate	2	**Duty cycle repetition rate** In Advance mode this is defined by inputs SPI_CLK and SPI_MOSI. See chapter 2.1.1. Otherwise efuse used: 1_D / 2_D Defines the time until next pulsing sequence starts in pulsing mode. 0_D: 250 µs (10km/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 Efuse used Defines the time sampling is active during one pulsing event. 0_D: 5 µs 1_D: 10 µs 2_D: 3 µs 3_D: 4 µs

Signal Name	Size	Function
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 & 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
mpa_ctrl	3	**Medium power amplifier gain control** Default after init sequence: 7_D / 5_D / 3_D / 1_D Efuse 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

3.2.11 Register reg8 – Divider

Table 29 **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 30 **Signal table of reg8**

Signal Name	Size	Function
div_sel	2	**Divider select** Default: 0_D Selects frequency divider setting. In default state internal 9.6MHz clock is selected. This clock is active only if SPImode with external clock is selected by QS1 input and the disable bit clk_ext_dis (reg15[13]) is not set. 0_D: Select internal 9.6MHz clock from oscillator 1_D: 2^13 2_D: 2^16 3_D: 2^20
div_out_en	1	**Divider out enable** Default: 0_D Enables the 2^13, 2^16, 2^20 divider logic. Does not affect setting 0 from div_sel, internal clock on pad `div_out` is enabled, if correspondent mode is selected with QS1 and Testmode is off (div_testmode_en=0). 0_D: Divider out off 1_D: Divider out on
div_testmode_en	1	**Enable divider testmode** Default: 0_D Puts VCO frequency divided by 16 on pad `div_out`. Overrides setting from bitfield div_sel. 0_D: Testmode off (div_sel active) 1_D: Testmode on

3.2.12 Register reg9 - Baseband

Table 31 **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

Table 32 **Signal table of reg9**

Signal Name	Size	Function
bb_hp_res	2	**High pass filter resistor settings** Default: 1_D / 3_D Efuse used 00_B: 8 MΩ 01_B: 4 MΩ 10_B: 2 MΩ 11_B: 1 MΩ
bb_clk_chop_sel	1	**Select clock chop frequency** Default: 1_D Selects frequency of clock for chopping (input for analog). In ABB it is divided by 2 (to get 50% duty cycle) and by 2 again. 0_D: 96 kHz 1_D: 192 kHz
bb_lpf_bw	1	**Low pass filter setting** Default: 0_D 0_D: 10 kHz 1_D: 60 kHz
bb_ctrl_gain	4	**Baseband PGA gain setting** Default: 40dB / 10dB / 25dB / 50dB (6_D / 0_D / 3_D / 8_D) Efuse 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

3.2.13 Register reg10 – Holdtime

Table 33 **Register assignment of reg10**

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

Reset value: 0x0000
Value after init sequence: Depends on QS3

Table 34 **Signal table of reg10**

Signal Name	Size	Function
hold	16	**Holdtime** Default after init sequence: Depends on QS3 Holdtime for target detection in steps of 128ms. However as the amplitude is filtered over 64 samples the shortest holdtime is 16/32/64/128ms depending on the PRT – and that minimum holdtime is selected by “0”.

3.2.14 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: Baseband 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

Signal Name	Size	Function
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

3.2.15 Register reg13 –Algo 1

Table 37 **Register assignment of reg13**

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

Reset value: 0x0000
Value after init sequence: 0x0000
Can be changed by metal patch

Table 38 **Signal table of reg13**

Signal Name	Size	Function
phase_win_len	3	**Phase Window Length** Default after init sequence: 0_D The phase difference is averaged during a window of this length. This setting is xor-ed with the dc_rep_rate before selection by the following table: 0_B: 256 1_B: 512 2_B: 256 3_B: 128 4_B: 64 5_B: 256 6_B: 256 7_B: 256
mean_win_len	3	**Mean Window Length** Default after init sequence: 0_D The dc-offset compensation averages during a window of this length. This setting is xor-ed with the dc_rep_rate before selection by the following table: 0_B: 256 1_B: 512 2_B: 256 3_B: 128 4_B: 64 5_B: 256 6_B: 256 7_B: 256
prt_mult	2	**Pulse Repetition Time Multiplier** Default after init sequence: 0_D If the repetition time is multiplied by the settings in reg2, the multiplier is configurable by the following table: 0_B: 4 1_B: 8 2_B: 16 3_B: 2

3.2.16 Register reg14 – Algo 2

Table 39 **Register assignment of reg14**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
thrs_offset								dir_hyst_dis	dir_keep	hold_x32	swap_iq	autoblind	pulse_mon	phase_thrs	

Reset value: 0x0000
Value after init sequence: 0x0000
Can be changed by metal patch

Table 40 **Signal table of reg14**

Signal Name	Size	Function
thrs_offset	8	**Threshold Offset** Default after init sequence: 0_D Possibility to shift the amplitude threshold up or down.
dir_hys_dis	1	**Direction Hysteresis disable** Default after init sequence: 0_D With this bit the direction detection can be tuned. 0_B: some hysteresis is used to switch between directions 1_B: no hysteresis is used and the default is “departing”
dir_keep	1	**Keep the Direction Algorithm Running** Default after init sequence: 0_D The behavior of the direction detection algorithm can be configured. 0_B: Only run while motion is detected. Otherwise output is “departing” 1_B: Keep the algorithm running during the hold-time even if no motion is detected.
hold_x32	1	**Multiply Hold-Time by 32** Default after init sequence: 0_D 0_B: No change 1_B: Holdtimes are longer by a factor of 32
swap_iq	1	**Swap I- and Q-Signal** Default after init sequence: 0_D 0_B: No change. 1_B: Swap IF signals for interpretation by direction algorithm. That leads to opposite direction detection.
autoblind_dis	1	**Disable Blanking Off Sensor (“dead-time”)** Default after init sequence: 0_D Disable blanking algorithm after detection. Do not change!

Signal Name	Size	Function
pulse_mon	1	**Monitor Radar Pulse** Default after init sequence: 0_D 0_B: Output pad PDet used as normal direction indication 1_B: Output pad PDet used to monitor internal radar pulse timing instead
phase_thrs	2	**Phase Threshold** Default after init sequence: 0_D Modify the threshold used by the direcition algorithm 0_B: No change (~5 degrees) 1_B: Divide by 2 2_B: Divide by 4 3_B: Set to 0

3.2.17 Register reg15 – Digital control

Table 41 **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	start_cw	fast_pahse	dir_c2_1		fastmode	adc_mom	miso_drv	mot_pol	dir_pol	stat_mux			

Reset value: 0x0000
Value after init sequence: 0x0000
Can be changed by metal patch

Table 42 **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. The reset transfers all FF into the power up state. Also the register itself is resetted, therefore writing 0 to this bit is not necessary. 0_B: Reset inactive 1_B: Reset active

Signal Name	Size	Function
start_pm	1	**Start pulsed mode** Default after init sequence: 0_D With this bit it is possible to start the pulsed mode (or CW mode) from SPI mode. Typical usecase is to configure registers and start the pulsed/CW mode afterwards by setting this bit to one. This is the only allowed usage of this bit. 0_B: Inactive 1_B: Rising edge triggers pulsed mode (or CW mode if bit 12 is set)
clk_ext_dis	1	**Disable external clock** Default after init sequence: 0_D In case the external clock is switched on by selecting the SPImode with external clock enabled, it can be switched off by setting this bit to 1. After switching off 16-32 further clock edges are delivered. 0_B: Clock not disabled 1_B: Clock disabled
start_cw	1	**Start CW mode instead** Default after init sequence: 0_D 0_B: No change 1_B: Changes behavior of bit 14 (“start_pm”) to start CW-mode instead of pulsed mode (both can be set in same SPI-access)
fast_phase	1	**Faster phase evaluation** Default after init sequence: 0_D 0_B: Start phase (direction) evaluation only when a target is detected. Therefore there is always some latency between Tdet and Pdet. 1_B: Start phase evaluation also before a detected target. Less latency, but higher risk of incorrect direction result. Much more difference in behavior can be achieved by setting bit 6 in reg14 to 1 also.
dir_c2_1	2	**Direction mode** Default after init sequence: 0_D Similar to bit “dir_mode” in reg2. Do not change!
fastmode	1	**SPI fast mode** Default after init sequence: 0_D 0_B: SPI_MISO (spi_do_o) changes on rising edge of SPI_CLK 1_B: SPI_MISO (spi_do_o) changes on falling edge of SPI_CLK
adc_mon	1	**ADC monitoring** Default after init sequence: 0_D For in-depth-debugging only. Do not change!

Signal Name	Size	Function
miso_drv	1	**SPI force MISO driver** Default after init sequence: 0_D 0_B: SPI_MISO (spi_do_o) is High-Z when SPI_CS_N=1 1_B: SPI_MISO (spi_do_o) is always driven to some level
mot_pol	1	**Motion polarity** Default after init sequence: 0_D 0_B: Tdet is low-active 1_B: Tdet is high-active
dir_pol	1	**Direction polarity** Default after init sequence: 0_D 0_B: Pdet is low when departing 1_B: Pdet is low when approaching
stat_mux	4	**Status multiplexer** Default after init sequence: 0_D For in-depth-debugging information only. Do not change! If set to anything !=0, the 10 MSB of reg56 are replaced by the following: 1_H: Advance-ADC thrs & "00" 2_H: Advance-ADC hold & "00" 3_H: Advance-ADC gnd & "00" 4_H: Advance-ADC vdd & "00" 5_H: "000000" & Advance & cs_n & mosi & clk 6_H: Amplitude 7_H: (Amplitude << 3) Others: qs1_s & init_done & qs2_s & qs3_s & qs4_s & Advance;

3.2.18 Register reg34 – ADC start

Table 43 **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 44 **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

3.2.19 Register reg35 – ADC convert

Table 45 **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 46 **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, fullscale analog input voltage 1.613V 1_D: gain = 1.00, fullscale analog input voltage 1.21 V

Signal Name	Size	Function
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.

3.2.20 Register reg36 – ADC Status

Table 47 **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 48 **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
Bg_up	1	**Bandgap_up** Default: 0_D This flag indicates if the bandgap is running. 0_D: Bandgap is not running or still booting 1_D: Bandgap running

3.2.21 Register reg38–53 – ADC result

Read only

These are the result registers of the ADC, a result is 10 bit wide, bit 0-9 of each register is occupied. Bits 10-15 are not used. As the ADC is physically an 8bit ADC also bit0 and bit1 are not used. Not used bits will deliver a zero when read.

Table 49 **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	QS2
7	45	QS3
8	46	Common mode voltage IFI
9	47	Common mode voltage IFQ
10	48	V_{DD} RF close to SPI
11	49	GND
12	50	Temperature sensor
13	51	PLL bandgap voltage
14	52	ADC bandgap voltage
15	53	ABB bandgap voltage

3.2.22 Register reg55 – E-fuses

Table 50 **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	Adaptive PRT	Direction output

Reset value: 0x0000

Read only

These bits influence the default value of corresponding bitfields of registers 4 – 9.

Table 51 **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
bb_hp_res	1	**Reg9 [7:6]** 0_D: bb_hp_res = 1_D 1_D: bb_hp_res = 3_D
Adaptive PRT	1	**Reg2 [14]** Power-saving option. Multiply PRT by 2/4/8/16 while no target is detected. Return to normal PRT after target detection for reliable direction measurement. 0_D: Standard PRT 1_D: Adaptive PRT

Signal Name	Size	Function
Direction output	1	**Reg2 [13]** 0_D: Direction default is departing when no motion is detected 1_D: Direction compatibility mode

3.2.23 Register reg56 – Status and chip version

Table 52 **Register assignment of reg56**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
quad_state1		init_done	quad_state2		quad_state3		quad_state4		Advance_mode			pll_lock_detect	chip_version		

Reset value: depending on chip_version and setting of “stat_mux” in reg15 (here the fields for “stat_mux”=0 is shown)
Value after init sequence: depending on chip_version and QS1, init_done=1, pll_lock_detect = 1

Read only

Table 53 **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 initial sequence after power up. 00_B: QS1 = GND 01_B: QS1 = open 10_B: QS1 = 100kΩ to VDD 11_B: QS1 = VDD
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
quad_state2	2	**Quad state input 2** These bits contain the read value from QS2 input which is read during initial sequence after power up. 00_B: QS2 = GND 01_B: QS2 = open 10_B: QS2 = 100kΩ to VDD 11_B: QS2 = VDD

Signal Name	Size	Function
quad_state3	2	**Quad state input 3** These bits contain the read value from QS3 input which is read during initial sequence after power up. 00_B: QS3 = GND 01_B: QS3 = open 10_B: QS3 = 100kΩ to VDD 11_B: QS3 = VDD
quad_state4	2	**Quad state input 4** These bits contain the read value from QS4 input which is read during initial sequence after power up. 00_B: QS4 = GND 01_B: QS4 = open 10_B: QS4 = 100kΩ to VDD 11_B: QS4 = VDD
Advance_mode	1	**Advance Mode Indicator** Default after init sequence: 1_D This bit reflects the sampled pll_trigger_i state. 0_D: Non-Advance mode 1_D: Advance mode
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 - here 3 Every variant has its own version number, it is hard wired on analog toplevel. These bits are read only.

3.2.24 Register GSR0 – SPI Status register

Table 54 **Register assignment of GSR0**

b7	b6	b5	b4	b3	b2	b1	b0
reserved					adc_result_ready	reserved	

Reset value: 0x0000

Read only

Table 55 **Signal table of GSR0**

The global status register GSR0 is sent on `spi_do_o` at the same time as the address and the read/write bit is sent on `spi_di_i`, MSB leading. There is only one bit used, it is bit 2 “adc_result_ready”. This is a flag for completed conversion. It is cleared by reading out any result register (reg38-reg53), adc clock has to be enabled for that.

4 Analog to digital converter

4.1 ADC conversion sequence

An ADC conversion consists of four different phases.

4.1.1 Enable bandgap

The bandgap is enabled by setting the bandgap_enable bit in register 34. This can be done simultaneously with clk_enable. The bandgap can be enabled or disabled independently of all other modules.

The startup time of the bandgap is temperature and device dependent. Enabling of the ADC is not allowed before the bg_up flag (reg36) is readout as high.

4.1.2 Enable local ADC clock

The local clock generator is enabled by setting the clk_enable bit in register 34, without setting any other bits, except bandgap_enable.

4.1.3 Enable ADC

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

Register 34 bit adc_enable, enables the ADC. The adc_ready bit high indicates a finished startup of the ADC. Conversion can not be started before adc_ready = ‘1’.

4.1.4 Start ADC conversion

4.1.4.1 Single conversion

A conversion is started by SPI write command into register 35, independent from the written data. During a running conversion, no further changes of these bits are allowed.

The ADC performs:

- start a sampling phase
- start a conversion phase
- update the corresponding result register
- set adc_result_ready bit to ‘1’

4.1.4.2 Sequential conversion

A conversion sequence for all input channels can be requested by writing register 35 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
- set adc_result_ready bit to ‘1’

The adc_result_ready bit within GSR0 is cleared by reading any of the result registers (register 38 – 53).

4.2 ADC configuration

4.2.1 Analog input channel gain

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

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

4.2.2 Analog input voltage sampling

During the first phase, the analog input voltage is sampled onto the DAC capacitor. This phase is called the sampling phase. The duration of this phase is controlled using the stc bits. Sampling time is fixed for ES, it is 16 clock cycles.

4.2.3 ADC phases

The physical resolution is 8 bit.

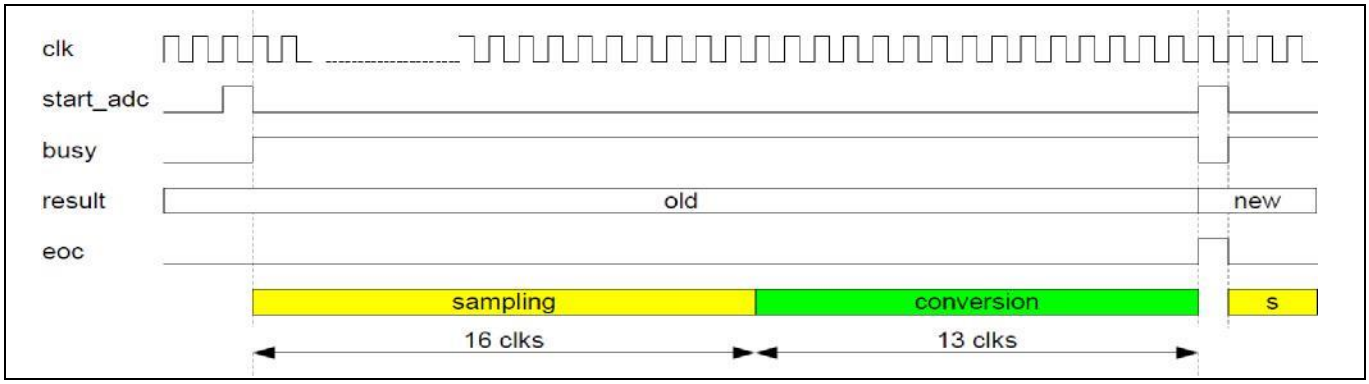

Figure 14 Timing diagram

4.3 Conversion Time

Example formula for calculation is provided below. 12 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 15MHz, max 50MHz).

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

$t_{conv\_min} = (t_{sample} + t_{distrib} + t_{epcal}) * t_{adc\_clk\_50M} = (16+13+12) * (1/50e6) = 0{,}82\mu s$

$t_{conv\_max} = (t_{sample} + t_{distrib} + t_{epcal}) * t_{adc\_clk\_15M} = (16+13+12) * (1/15e6) = 2{,}73\mu s$

4.4 ADC power-down sequence

In case a low current consumption mode is required a full ADC power-down can be invoked in 2 phases:

1) Disable ADC by setting adc_enable to ‘0’. The clock must still be running to enable the FSMs 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.

5 Detector

5.1 Digital evaluation

The detector is responsible for evaluating the input from the ADC and for setting of Tdet/Pdet outputs of the BGT60LTR11AIP.

Target detected (Tdet) ouput is low active. Phase detected (Pdet) output is used to show the direction of the detected target. It is set high for approaching targets, otherwise low.

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

5.1.1 Holdtime

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

It can be configured in reg10.

Revision history

Document version	Date of release	Description of changes
V1.0	2020-09-09	First preliminary release
V1.1	2020-10-06	Added autonomous mode
V1.2	2021-07-15	Changes all over the document

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

Edition 2021-07-15

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.