User guide to BGT60LTR11AIP

XENSIV™ 60 GHz radar

About this document

Scope and purpose

This user guide is intended to provide more details on how to use BGT60LTR11AIP in an actual user guide in addition to the datasheet.

Since the datasheet gives only technical data and limits of the device itself, this user guide is explaining how to operate the device in greater detail, and describes:

- All different building blocks
- How to operate the different blocks
- Settings of the SPI registers are grouped on topic, including truth tables

Intended audience

This document serves as a primer for firmware or software engineers who want to get started with hardware design for Infineon's XENSIV™ 60 GHz BGT60LTR11AIP and its derivatives BGT60LTR11SAIP and BGT60LTR11BAIP.

Table of contents

About this document . . . 1
Table of contents . . . 2
1 Introduction . . . 4
2 Main controller . . . 5
2.1 Quad state inputs and “Advance mode” . . . 6
2.1.1 Quad state basics . . . 6
2.1.2 “Advance mode” . . . 6
2.1.3 QS1 . . . 6
2.1.4 QS2 . . . 7
2.1.5 QS3 . . . 8
2.1.6 QS4 . . . 9
2.2 Power-up and sequencing . . . 9
2.2.1 Power-up . . . 10
2.2.2 Init sequence . . . 10
2.2.3 Autonomous pulsed mode sequence . . . 12
2.2.4 Autonomous CW mode sequence . . . 15
2.2.5 SPI mode sequence . . . 16
2.2.6 Detector . . . 16
2.3 Overview dynamic control signals . . . 16
3 SPI interface . . . 18
3.1 SPI interface description . . . 19
3.1.1 SPI write mode . . . 19
3.1.2 SPI read mode . . . 19
3.1.3 SPI burst mode . . . 20
3.2 SPI registers . . . 21
3.2.1 Register overview . . . 21
3.2.2 Direct access register . . . 22
3.2.3 Register map bitfields . . . 22
3.2.4 Register Reg0 – Direct access register . . . 24
3.2.5 Register Reg1 – Direct access register . . . 26
3.2.6 Register Reg2 – Threshold . . . 27
3.2.7 Register Reg4 – PLL config 1 . . . 28
3.2.8 Register Reg5 – PLL config 2 . . . 30
3.2.9 Register Reg6 – PLL config 3 . . . 31
3.2.10 Register Reg7 – Duty cycling, timing, PD, MPA . . . 32
3.2.11 Register Reg8 – Divider . . . 34
3.2.12 Register Reg9 – Baseband . . . 35
3.2.13 Register Reg10 – Hold time . . . 37

Table of contents

3.2.14 Register Reg12 – BITE . . . 37
3.2.15 Register Reg13 – Algo 1 . . . 38
3.2.16 Register Reg14 – Algo 2 . . . 40
3.2.17 Register Reg15 – Digital control . . . 41
3.2.18 Register Reg34 – ADC start . . . 43
3.2.19 Register Reg35 – ADC convert . . . 44
3.2.20 Register Reg36 – ADC status . . . 44
3.2.21 Register Reg38–53 – ADC result . . . 46
3.2.22 Register Reg56 – Status and chip version . . . 47
3.2.23 Register GSR0 – SPI status register . . . 49

4 Analog to digital converter . . . 50
4.1 ADC conversion sequence . . . 50
4.1.1 Enable bandgap . . . 50
4.1.2 Enable local ADC clock . . . 50
4.1.3 Enable ADC . . . 50
4.1.4 Start ADC conversion . . . 50
4.1.4.1 Single conversion . . . 50
4.1.4.2 Sequential conversion . . . 50
4.2 ADC configuration . . . 51
4.2.1 Analog input channel gain . . . 51
4.2.2 Analog input voltage sampling . . . 51
4.2.3 ADC phases . . . 51
4.3 Conversion time . . . 51
4.4 ADC power-down sequence . . . 52

5 Detector . . . 53
5.1 Digital evaluation . . . 53
5.2 Hold time . . . 53

Revision history . . . 54

Disclaimer . . . 55

1 Introduction

The BGT60LTR11AIP is a fully integrated microwave motion sensor including antenna structures, configurable built-in detector and a state machine allowing for fully autonomous operation. It is designed to operate as a Doppler motion sensor in the frequency band from 61 GHz to 61.5 GHz for the BGT60LTR11AIP and BGT60LTR11SAIP versions, and from 60.5 GHz to 61 GHz for the BGT60LTR11BAIP version.

An integrated frequency divider with a phase locked loop (PLL) provides a voltage-controlled oscillator (VCO) frequency stabilization and allows for continuous wave (CW) operation. The device supports two operation modes, fully 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 detector. The packaged monolithic microwave integrated circuit (MMIC) features integrated broad beam antennas for maximum area coverage.

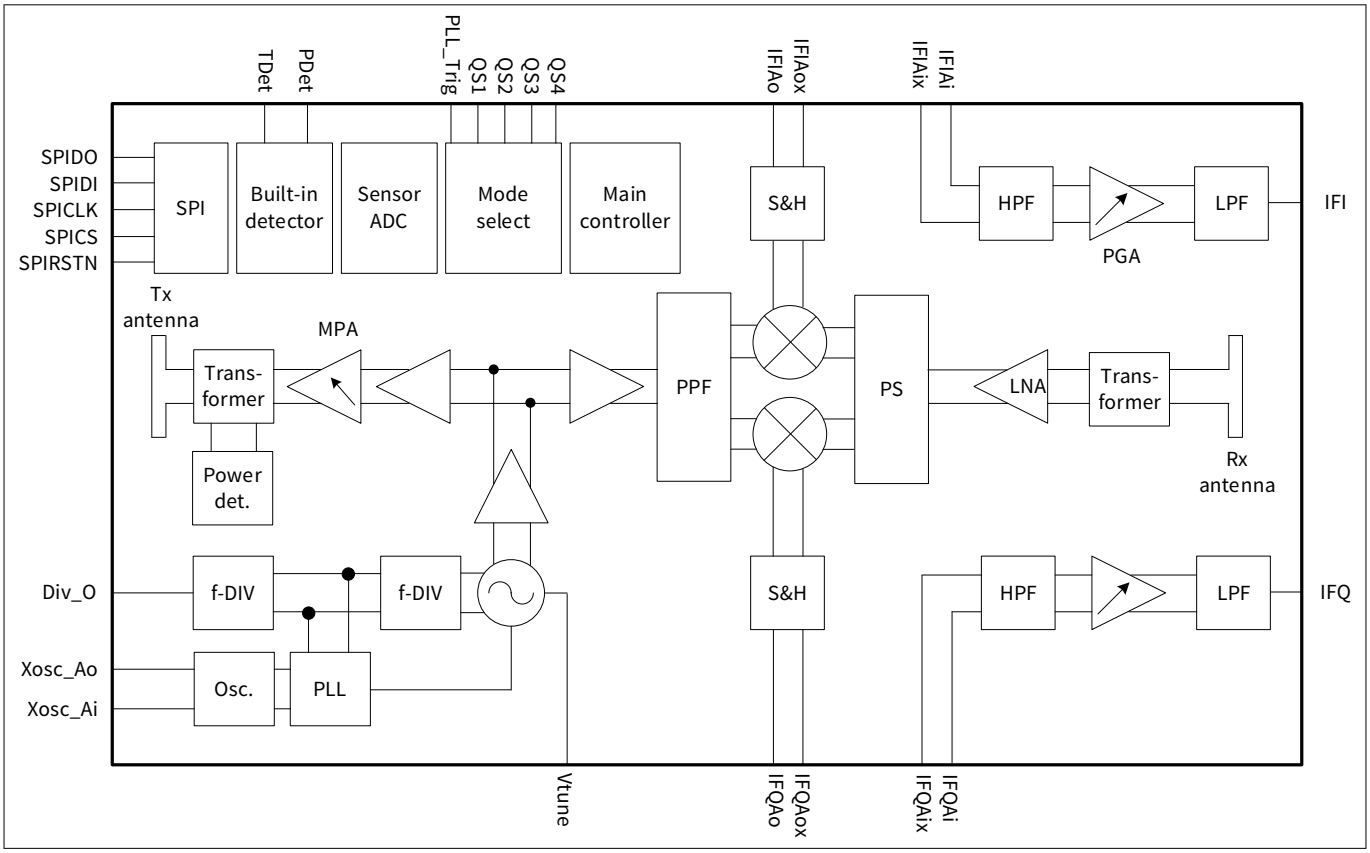


Figure 1 BGT60LTR11AIP block diagram

2 Main controller

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

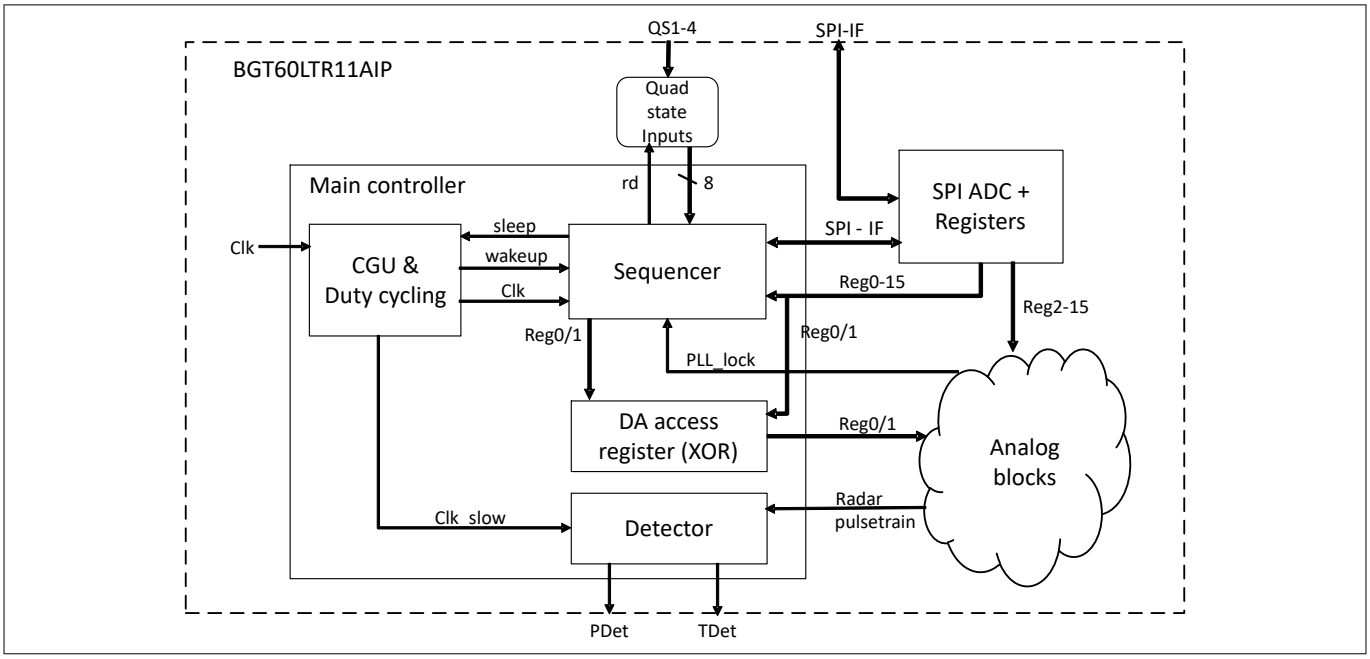


Figure 2 Main controller block diagram

The default direction information is set to “approaching”. The truth table for the output pad voltage is shown in Table 1.

Table 1 Truth table for the output pad voltage

Output pad voltage		Motion	Approaching/Departing
TDet	PDet		
high	high	No	N.A.
low	low	Yes	Departing
low	high	Yes	Approaching

The 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.

2.1 Quad state inputs and “Advance mode”

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

2.1.1 Quad state basics

The quad state inputs allow configuring four different states with one input pin. Table 2 shows possible input states and the respective 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 soft_reset bit (Reg15[15]).

Table 2 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

2.1.2 “Advance mode”

When the PLL_Trig pin is "0" during the digital main controller wakes up from reset, the chip boots in "Basic mode".

The "Advance mode" is enabled when the PLL_Trig pin is kept "1" during chip boot and QS1 is either GND or OPEN, pins SPIDI and SPICLK are also sampled to determine the pulse repetition time (PRT): dc_rep_rate (Reg7[11:10]). In addition, pins QS2 and QS3 are evaluated by the sensor ADC and converted in 4-bit values each before each “mean window”.

Table 3 PRT in Advance mode

PLL_Trig	SPIDI	SPICLK	dc_rep_rate	PRT
0	*	*	1	500 µs
1	0	0	1	500 µs
1	0	1	3	2000 µs
1	1	0	0	250 µs
1	1	1	2	1000 µs

2.1.3 QS1

QS1 is used to select the mode of the chip (Autonomous or SPI).

Table 4 QS1

Pad	b1	b0	Operating mode
ground	0	0	Autonomous CW mode[1)]

(table continues...)

Table 4 (continued) QS1

Pad	b1	b0	Operating 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) Not a normal working mode. Only for test purpose, e.g. FCC, ETSI.

2.1.4 QS2

QS2 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 Reg2 described in Register Reg2 – Threshold.

Table 5 QS2

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

1) Detector threshold set values ensure a 0.5-m detection range step.
2) Assigned QS state according to the sampled voltage range: e.g., $1*V_{DD}/16 – 2*V_{DD}/16$ means $1*V_{DD}/16$ till $2*V_{DD}/16$.

2.1.5 QS3

QS3 is used to select the hold time of the TDet output, this defines how long the output status will be kept after a target is detected, it is written into register Reg10 described in Register Reg10 – Hold time.

Table 6 QS3

Pad	b1	b0	Detector hold time
Basic mode			
ground	0	0	16/32/64/128 ms (dep. on dc_rep_rate)
open	0	1	1 s
100 kΩ to V_{DD}	1	0	10 s
V_{DD}	1	1	1 min
Advance mode			
$0*V_{DD}/16 – 1*V_{DD}/16$[1)]	*	*	100 ms
$1*V_{DD}/16 – 2*V_{DD}/16$	*	*	500 ms
$2*V_{DD}/16 – 3*V_{DD}/16$	*	*	1 s
$3*V_{DD}/16 – 4*V_{DD}/16$	*	*	2 s
$4*V_{DD}/16 – 5*V_{DD}/16$	*	*	3 s
$5*V_{DD}/16 – 6*V_{DD}/16$	*	*	5 s
$6*V_{DD}/16 – 7*V_{DD}/16$	*	*	10 s
$7*V_{DD}/16 – 8*V_{DD}/16$	*	*	30 s
$8*V_{DD}/16 – 9*V_{DD}/16$	*	*	45 s
$9*V_{DD}/16 – 10*V_{DD}/16$	*	*	1 min
$10*V_{DD}/16 – 11*V_{DD}/16$	*	*	90 s
$11*V_{DD}/16 – 12*V_{DD}/16$	*	*	2 min
$12*V_{DD}/16 – 13*V_{DD}/16$	*	*	5 min
$13*V_{DD}/16 – 14*V_{DD}/16$	*	*	10 min
$14*V_{DD}/16 – 15*V_{DD}/16$	*	*	15 min
$15*V_{DD}/16 – 16*V_{DD}/16$	*	*	30 min

1) Assigned QS state according to the sampled voltage range: e.g., $1*V_{DD}/16 – 2*V_{DD}/16$ means $1*V_{DD}/16$ till $2*V_{DD}/16$.

2.1.6 QS4

QS4 is used to select the device operating frequency by configuring the PLL. Frequency is also dependent on version of the chip: BGT60LTR11AIP and BGT60LTR11SAIP, or BGT60LTR11BAIP.

Table 7 QS4

Pad	b1	b0	VCO frequency
BGT60LTR11AIP and BGT60LTR11SAIP			
ground	0	0	61.1 GHz
open	0	1	61.2 GHz
100 kΩ to V_{DD}	1	0	61.3 GHz
V_{DD}	1	1	61.4 GHz
BGT60LTR11BAIP			
ground	0	0	60.6 GHz
open	0	1	60.7 GHz
100 kΩ to V_{DD}	1	0	60.8 GHz
V_{DD}	1	1	60.9 GHz

Warning: ***Sensors operating in close vicinity at the same operating frequency can interfere!***

2.2 Power-up and sequencing

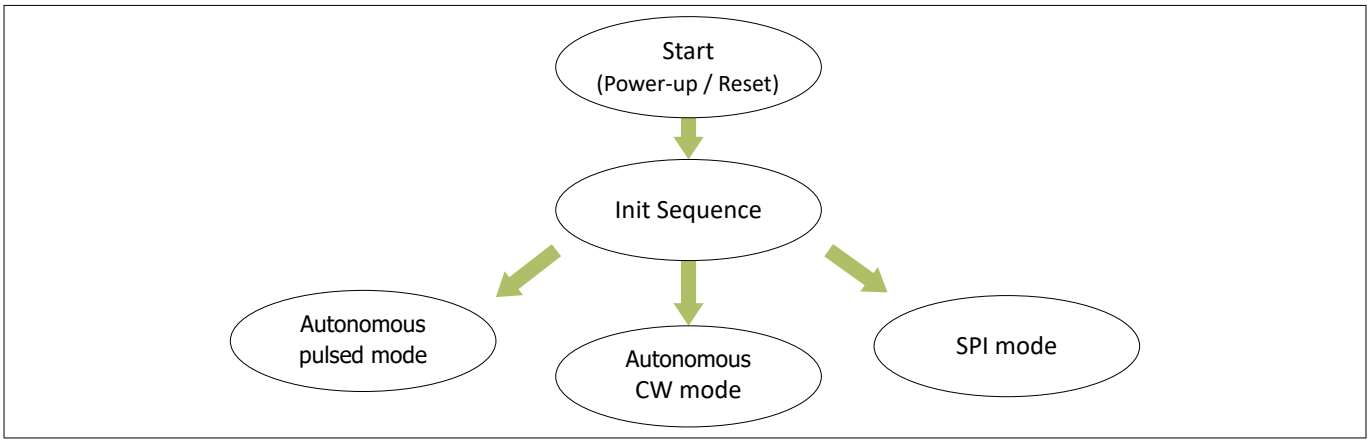


Figure 3 State diagram overview

Depending on quad state input 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 autonomous (pulsed/CW) mode. So SPI activity is recommended only while in SPI mode.

The reason is that the main controller is halted as long as the pad SPICSN 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.

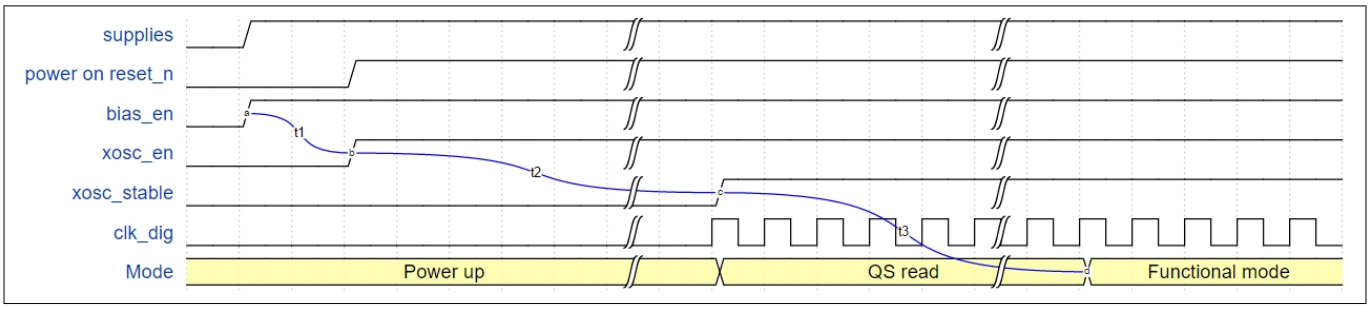


Figure 4 Power-up

Supply ramp needs to be shorter than 400 µs. The 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 µs. 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, and in SPI (pulsed/CW) mode, it takes 25 µs more.

2.2.2 Init sequence

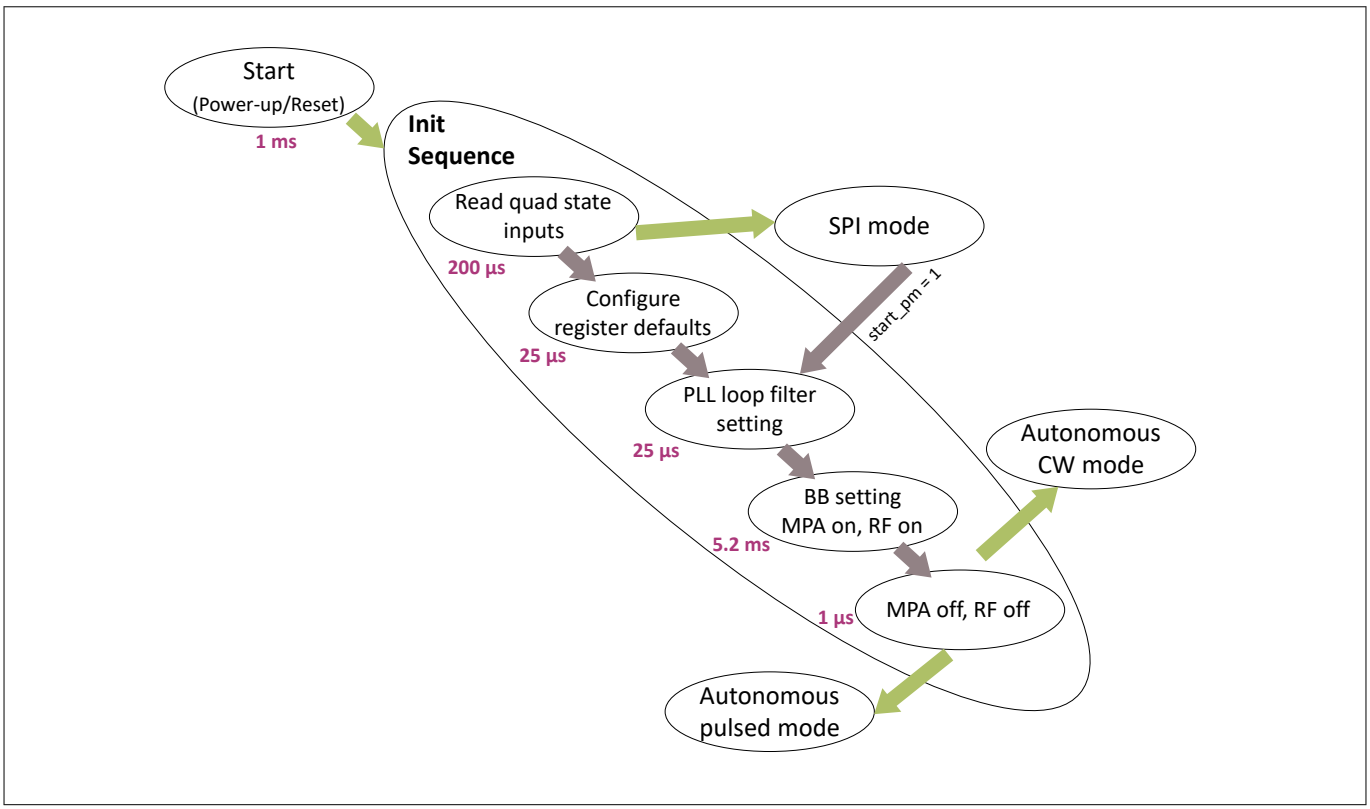


Figure 5 Init sequence

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

1. Read quad state inputs, which needs 200 µs for analog settling, to know the selected operation mode. If it is SPI mode, the init sequence is terminated, then the main controller switches to SPI mode and the control over the SPI interface is handed over to an external microcontroller
2. If it is autonomous (pulsed/CW) mode, the default values for the configuration registers are prepared and written using the SPI interface of the SPI ADC block internally
3. PLL starts, medium power amplifier (MPA) is not activated during the loop filter settlement. After 20 µs also the MPA is activated, RF is also running. The next 5 ms are used for baseband (BB) settling
4. For autonomous CW mode, the init sequence is terminated and the main controller switches to CW mode
5. For autonomous pulsed mode, MPA, RF and PLL is switched off and the main controller switches to pulsed mode

The status bit init_done (Reg56[13]) is set when leaving init sequence.

Table 8 Init sequence in detail

Nr	Command	Description
1	write Reg1 0x0100	Set bit qs_rd_en
2	2*read Reg55	Read predefined values twice to provide enough clock cycles
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
7	spiwrite Reg5 Reg5_init	Write defined default value
8	spiwrite Reg6 Reg6_init	Write defined default value
9	spiwrite Reg7 Reg7_init	Write defined default value (partly calculated from pin PLL_Trig, SPICLK and SPIDI)
10	spiwrite Reg8 Reg8_init	Write defined default value
11	spiwrite Reg9 Reg9_init	Write defined default value
12	spiwrite Reg2 Reg2_init	Write defined default value (calculated from pin PLL_Trig, and quad states – analog read-in in Advance mode)
13	spiwrite Reg10 Reg10_init	Write defined default value (calculated from pin PLL_Trig, and quad states – analog read-in in Advance mode)
14	write Reg0 0x311F	Set vcobuf_en, vco_en, pll_en, rxbuf_en, txbuf_en, mixi_en, mixq_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 to 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 to sample enable	Wait defined time MPA enable to Sample&Hold – mpa2sh_dly (Reg7[5:4])

(table continues...)

Table 8 (continued) Init sequence in detail

Nr	Command	Description
22	write Reg1 0x1037	Set bb_sample_en
23	wait 5 ms	Wait time for baseband settling
24	end for autonomous CW mode	End init sequence if mode = autonomous CW mode
25	write Reg1 0x1092	Reset bb_boost_dis, bb_strup_hp, bb_sample_en, set bb_dig_det_en Wait 100 ns
26	write Reg0 0x371F	Reset mpa_en
27	wait 20 µs	Allow settling of PLL without active MPA for best re-locking in pulsed mode
28	write Reg0 0x311F	Reset pll_active, pll_clk_gate_en Wait 100 ns
29	write Reg1 0x0092	Reset div_bias_en
30	write Reg0 0x0900	RF off, only pll_en is still on and pll_open_loop is set
31	end for autonomous pulsed mode	End init sequence for pulsed mode, pulsed mode starts with sleep phase defined by dc_rep_rate (Reg7[11:10]). Afterwards the autonomous pulsed mode sequence is started

2.2.3 Autonomous pulsed mode 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 500 µs and an on time of 14 µs which consists of about 9 µs needed for RF and PLL startup, and 5 µs as default sample time.

Table 9 Autonomous pulsed mode sequence in detail

	Command	Description
1	write Reg0 0x391F	Set vcobuf_en, vco_en, pll_en, rxbuf_en, txbuf_en, mixi_en, mixq_en, lna_en
2	write Reg1 0x1092	Set div_bias_en
3	wait VCO to 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 to 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 100 ns

(table continues...)

Table 9 (continued) Autonomous pulsed mode sequence in detail

	Command	Description
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 wake-up	Sequence is restarted after defined time. The dc_rep_rate (Reg7[11:10]), dc_rep_rate measures from active phase to active phase

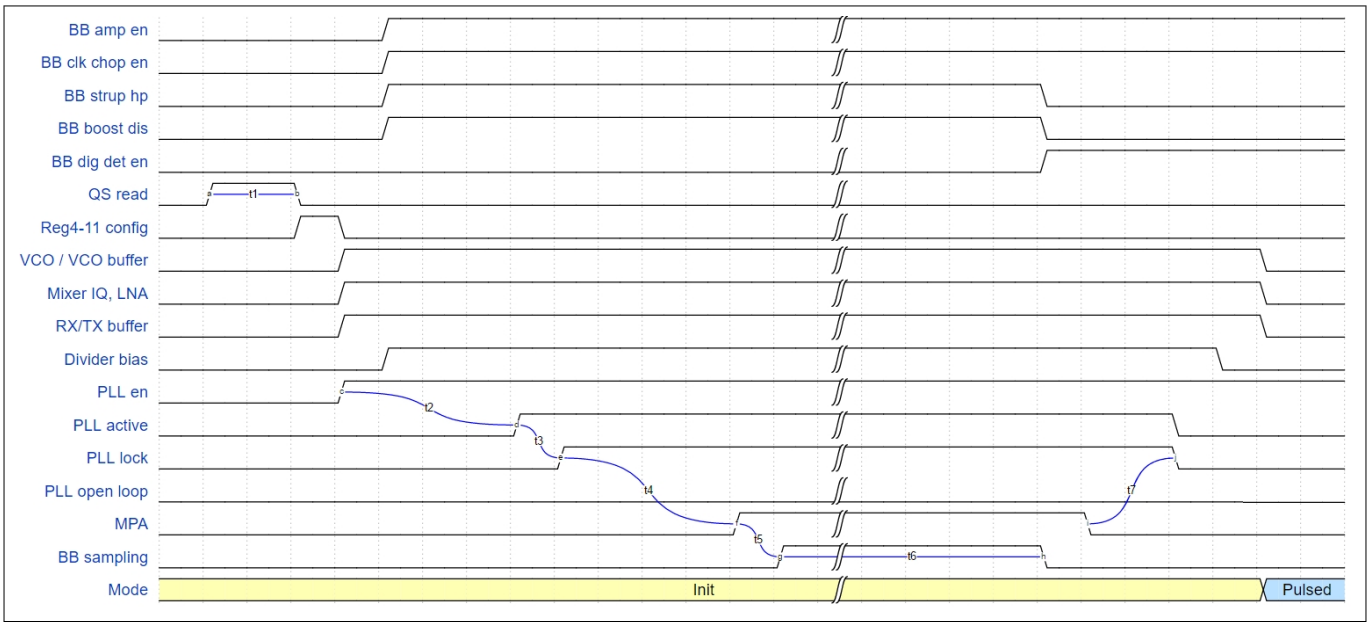


Figure 6 Autonomous pulsed mode – init phase

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

t1: 200 µs, time required for quad states read input to be one before reading them

Before switching on any block, the config registers Reg4-11 are written.

t2: 2 µs, time required for PLL enable to be enabled before PLL is set into active state. Shortly after switching on of the VCO, divider bias has to be activated

t3: ~15 µs, time required for PLL locking

t4: 20 µs, time required for loop filter settling with MPA disabled

t5: mpa2sh_dly (1 µs default), time between activating MPA and sampling

t6: 5 ms, time required for baseband settling

t7: 20 µs, time required for PLL settling with MPA disabled to allow best re-locking condition in pulsed mode

Now 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 pulsed mode at the same time.

For the init phase, PLL and RF are switched off in the following order: Sample&Hold / MPA / PLL / Divider bias / RF, each step has to take 100 ns. A time t_7 is required after MPA is disabled. 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. The bb_clock_chop, digital detector and wake-up 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 wake-up pulse from wake-up 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 wake-up will happen with the next wake-up pulse

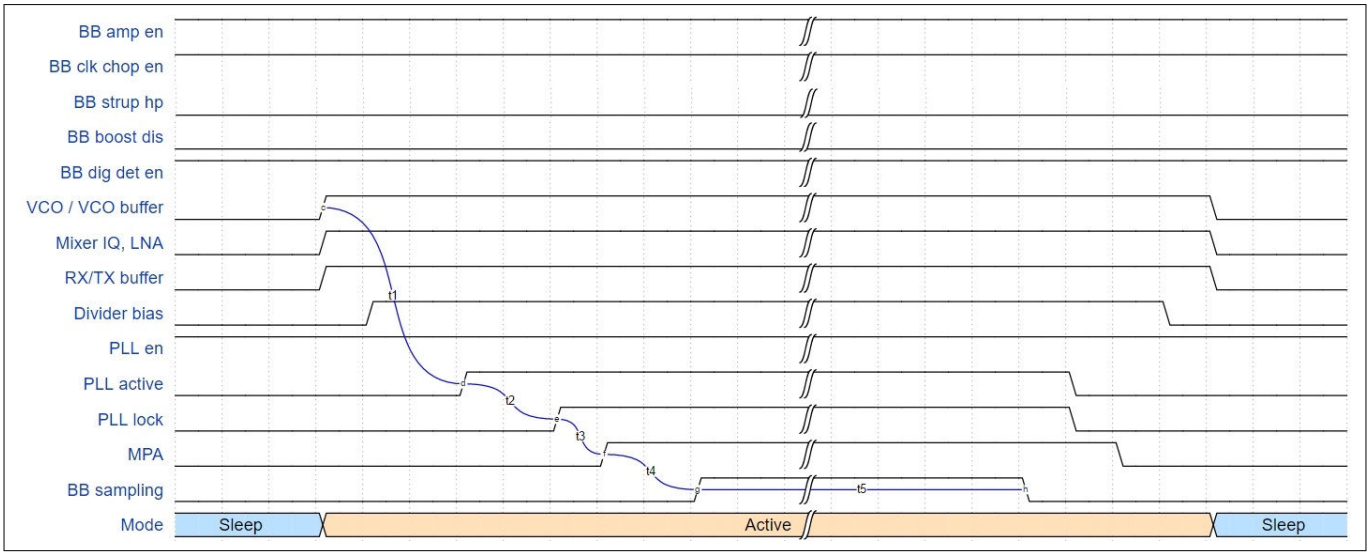


Figure 7 Pulsed mode – active phase

Figure 7 shows the pulsed mode after the init phase starting with a sleep phase and also ending with a sleep phase. The dc_rep_rate (Reg7[11:10]) defines the time from start of active phase until start of next active phase.

t1: 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

t2: ~5 µs, time PLL needs for locking

t3: 300-400 ns, time needed for synchronizing lock detect signal and enabling MPA

t4: mpa2sh_dly (1 µs default), time between activating MPA and sampling

t5: dc_on_pulse_len (5 µs default) Reg7[9:8], sampling on time

For the active phase, PLL and RF are switched off in the following order: Sample&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.

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

2.2.4 Autonomous CW mode sequence

In autonomous continuous wave mode, the device is active as configured with quad state inputs. The main controller only uses the sensor ADC and evaluates the IF signals in the digital detector. The init sequence was left at the right step, so the CW mode 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 10 Autonomous CW mode sequence in detail

	Command	Description
1	write Reg1 0x10B3	Set bb_dig_det_en, reset bb_strup_hp
2	end	

Figure 8 shows autonomous CW mode including init phase.

Init phase until end settling time for baseband is the same as for autonomous pulsed mode. As in CW mode, baseband and sampling is on continuously, bb_strup_hp=0 (Reg1[2]).

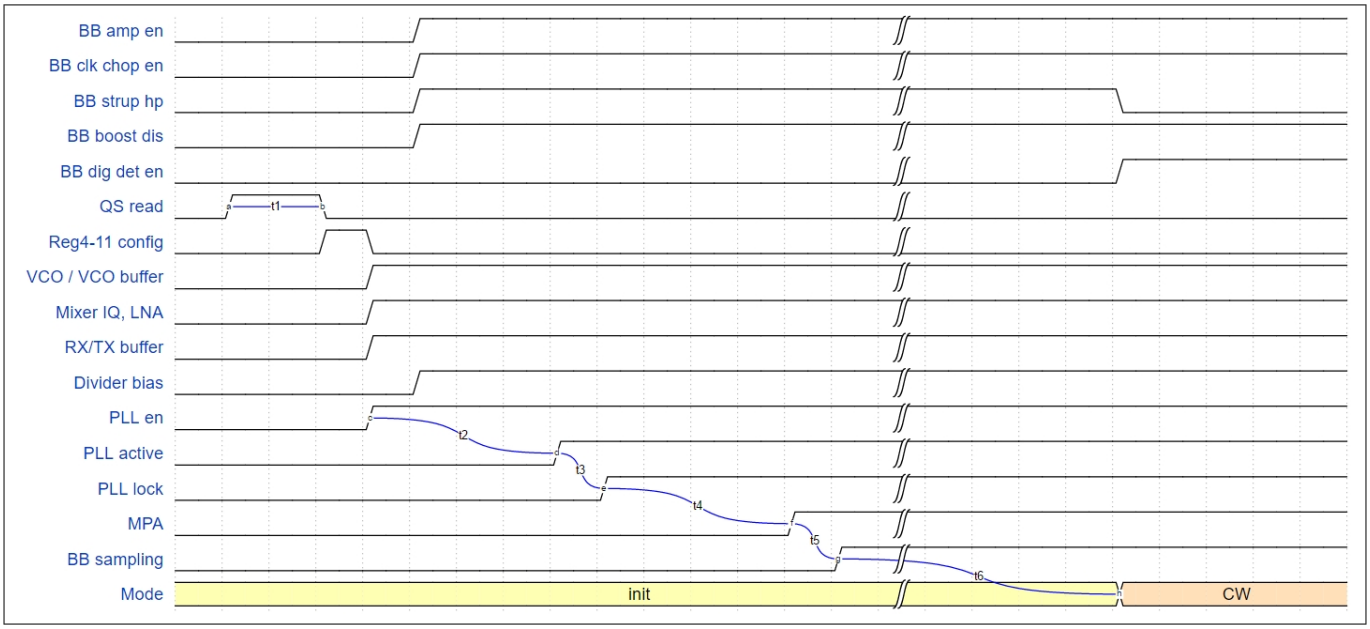


Figure 8 Autonomous CW mode

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 QS1, external 9.6 MHz clock is enabled (QS1="10") or not (QS1="11").

Table 11 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 start_cw=0 (Reg15[12]), 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, the detector can be activated. It takes another 50 ms 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 12 Overview dynamic control signals

	Autonomous pulsed mode	Autonomous 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
rxbuf_en, 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 100 ns 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 the main controller continuously. Programming it with “1” would just invert the bit

3 SPI interface

SPI – Serial Peripheral Interface

- 7-bit continues address space
- Fixed payload of 16-bits
- 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 SPICS=1 the data output SPIDO (a.k.a 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 SPICSN high. Otherwise the internal sequencer cannot continue and operation stops
- While the chip is working in autonomous mode (pulsed/CW), the following timing constraints must be considered. **In order to make SPIDO available, it is highly recommended to set bit “miso_drv” (Reg15[6])!**
  - During accesses, SPICSN must not be driven low too long (1.6 µs). That would prevent the internal use of SPI and break the timing in the sequencer. This is mostly relevant with low data rates (below 16 Mbps) or with burst accesses
  - When starting an access, the following arbitration protocol must be used:
    1. Initially SPICSN is driven high. Minimum for ≥100 ns
    2. Wait until SPIDO 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 SPICSN low (try to reserve the bus)
    4. Wait for ≥100 ns (the time needed for synchronization of SPICSN)
    5. Check SPIDO again. If it is high now (sequencer has just started SPI also), go to step 1
       - Otherwise, continue to step 6
    6. Do the actual data transmission
    7. Drive SPICSN high (release the bus)
- As an alternative to the restrictive timing and arbitration scheme during active mode above, it is also possible to use the SPI during the slots when it is not used internally. That means watching the periodic pattern (period is the sample rate) on SPIDO and align external accesses accordingly

3.1 SPI interface description

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

The working edge is the rising edge of the clock. The status of SPIDI is shifted into the input register with every working edge. Also, with every working edge, the state of the SPIDO is shifted out of the output. This timing on SPIDO can be changed by setting fast_mode (Reg15[8]) to "1". In that case, SPIDO is delayed by half of an 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 SPICSN with transfer of the 7-bit address, MSB first. The followed 8th 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 address and RW bit are received, the global status register GSR0 (8-bit) is serially shifted out on SPIDO (also MSB first). While sending the payload, the previous register content is serially shifted out on SPIDO.

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

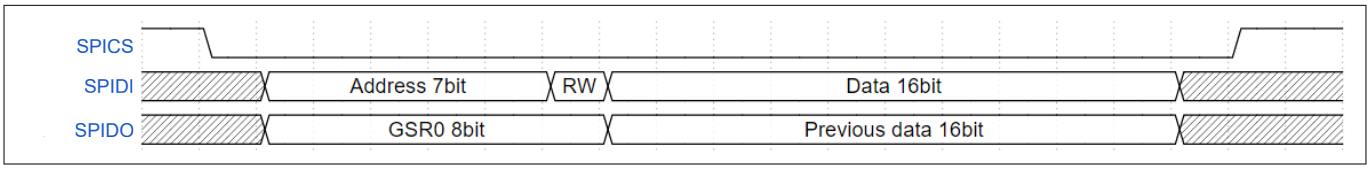


Figure 9 SPI write – MSB first anytime, RW="1"

3.1.2 SPI read mode

A read access starts after the falling edge of SPICSN with transfer of the 7-bit address, MSB first. The followed 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 SPIDO (also MSB first). Directly after that the read data is serially shifted out on SPIDO.

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

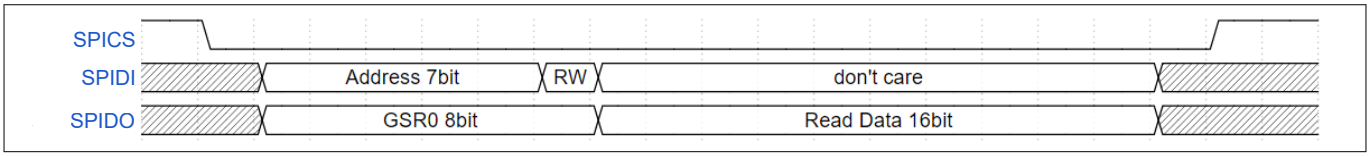


Figure 10 SPI read – MSB first anytime, RW="0"

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 13.

Burst command examples:

Burst command to access register Reg4 to read: 0xFF08, and to write: 0xFF09.

Table 13 Burst mode command

Bit field	Bit width	Bit field name	Description
15:8	8	addr/RW	Address to start the burst (always 0xFF)
7:1	7	saddr	Burst mode starting address
0	1	rwb	Burst mode read/write 0 - burst read 1 - burst write

SPI burst access

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

Burst mode read sequence

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

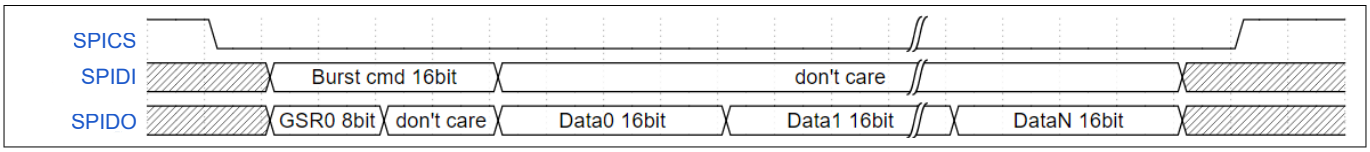


Figure 11 SPI burst read

Burst mode write sequence

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

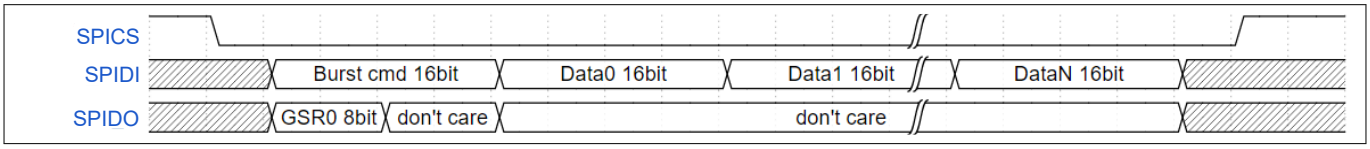


Figure 12 SPI burst write

3.2 SPI registers

3.2.1 Register overview

Table 14 Register overview

Register	Mode	Contents	Reset value	Value after init sequence (pulsed sleep/CW)
Reg0	RW	Control bits	0x0000	0x0900[1] / 0x373F[1]
Reg1	RW	Control bits	0x0000	0x0092[1] / 0x10B3[1]
Reg2	RW	Detector threshold	0x0000	dep. on QS2
Reg3	RW	Reserved	0x0000	0x0000
Reg4	RW	PLL 1	0x0000	0x0F3A
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	0x0068
Reg10	RW	Detector hold time	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
Reg56	RO	Status and chip version	0x0000	0x2000, Bits [0:2] dep. on chip_version Bits [7:8] dep. on QS4 Bits [9:10] dep. on QS3 Bits [11:11] dep. on QS2 Bits [15:14] dep. on QS1
GSR0	RO	8-bit SPI status register	0x00	0x00

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

The “real” MMIC reset values can be read only in SPI mode after reset, as they are not changed in this mode. In autonomous mode (pulsed/CW), the MMIC reset values are overwritten when the init sequence has finished.

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 need to be programmed by SPI from external optional microcontroller or internally from the main controller at power-up.

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.

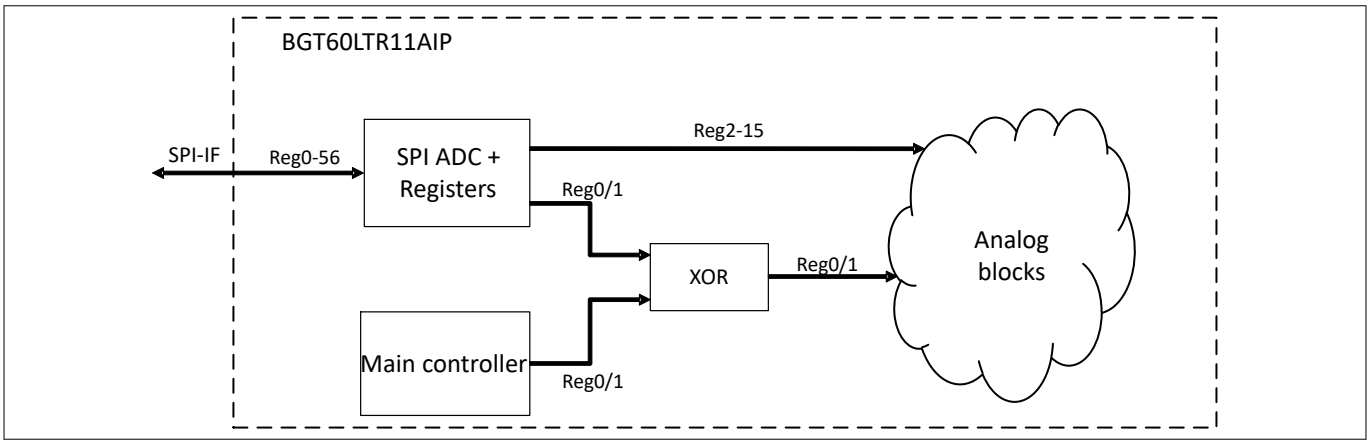


Figure 13 Direct access registers

3.2.3 Register map bitfields

Table 15 Register map

		7	6	5	4	3	2	1	0
Reg0	**[15:8]**	reserved		vcobuf_en	vco_en	pll_open_loop	pll_clk_gate_en	pll_active	pll_en
	[7:0]	reserved		mpa_en	txbuf_en	mixq_en	mixi_en	lna_en	rxbuf_en
Reg1	**[15:8]**	reserved			div_bias_en	reserved			qs_rd_en
	[7:0]	bb_dig_det_en	reserved	bb_boost_dis	bb_clk_chop_en	reserved	bb_strup_hp	bb_amp_en	bb_sample_en
Reg2	**[15:8]**	hprt	aprt	dir_mode	thrs				
	[7:0]	thrs							
Reg3	**[15:8]**	reserved							
	[7:0]								
Reg4	**[15:8]**	pll_dft_dmux		reserved	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			pll_cp_mode	pll_pfd_rdt_sel	

(table continues...)

Table 15 (continued) Register map

		7	6	5	4	3	2	1	0
Reg5	**[15:8]**	reserved				pll_fcw			
	[7:0]	pll_fcw							
Reg6	**[15:8]**	pll_ld_tw_sel			pll_ld_len	pll_ld_en	reserved		
	[7:0]	reserved							
Reg7	**[15:8]**	reserved				dc_rep_rate		dc_on_pulse_len	
	[7:0]	reserved	vco2pll_dly	mpa2sh_dly		pd_en	mpa_ctrl		
Reg8	**[15:8]**	reserved							
	[7:0]	reserved				div_sel		div_out_en	div_testmode_en
Reg9	**[15:8]**	reserved							
	[7:0]	bb_hp_res		bb_clk_chop_sel	bb_lpf_bw	bb_ctrl_gain			
Reg10	**[15:8]**	hold							
	[7:0]								
Reg11	**[15:8]**	reserved							
	[7:0]								
Reg12	**[15:8]**	reserved							
	[7:0]	bb_amux_ctrl		bb_amux_en	bite_pd_en	bite_ctrl			bite_en
Reg13	**[15:8]**	reserved							
	[7:0]	phase_win_len			mean_win_len			prt_mult	
Reg14	**[15:8]**	thrs_offset							
	[7:0]	dir_hys_dis	dir_keep	hold_x32	swap_iq	autoblind_dis	pulse_mon	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		chnr_all	chnr			

(table continues...)

Table 15 (continued) Register map

		7	6	5	4	3	2	1	0
Reg36	**[15:8]**	reserved							
	[7:0]	reserved						adc_ready	bandgap_up
Reg38 - Reg53	**[15:0]**	ADC result register channel 0 - 15							
Reg56	**[15:8]**	qs1_s		init_done	qs2_s		qs3_s		qs4_s
	[7:0]	qs4_s	advance_mode	reserved		pll_lock_detect	chip_version		
GSR0	**[7:0]**	reserved					adc_result_ready	reserved	

3.2.4 Register Reg0 – Direct access register

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

DAR_REG0_REG **Address:** $0x00_H$

Register assignment of Reg0 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res		**vcobuf_en**	**vco_en**	**pll_open_loop**	**pll_clk_gate_en**	**pll_active**	**pll_en**
rw		rw	rw	rw	rw	rw	rw

7	6	5	4	3	2	1	0
Res		**mpa_en**	**txbuf_en**	**mixq_en**	**mixi_en**	**lna_en**	**rxbuf_en**
rw		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Res	15:14	rw	**Not Used. Do not change reset values.**
vcobuf_en	13	rw	**Enable VCO buffer** 0_D: VCO buffer off 1_D: VCO buffer on
vco_en	12	rw	**Enable VCO** 0_D: VCO off 1_D: VCO on
pll_open_loop	11	rw	**Enable PLL open loop clock gating** Switches PLL into open loop after lock detect was reached 0_D: closed loop 1_D: open loop after lock detect

(table continues...)

(continued)

Field	Bits	Type	Description
pll_clk_gate_en	10	rw	**Enable PLL clock gating** 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	9	rw	**PLL active** PLL locking is started when this bit is set 0_D: PLL loop open 1_D: PLL locking started
pll_en	8	rw	**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
Res	7:6	rw	**Not Used. Do not change reset values.**
mpa_en	5	rw	**Enable Medium Power Amplifier** 0_D: MPA off 1_D: MPA on
txbuf_en	4	rw	**Enable TX buffer** 0_D: TX buffer off 1_D: TX buffer on
mixq_en	3	rw	**Enable Mixer Q** 0_D: Mixer Q off 1_D: Mixer Q on
mixi_en	2	rw	**Enable Mixer I** 0_D: Mixer I off 1_D: Mixer I on
lna_en	1	rw	**Enable LNA** 0_D: LNA off 1_D: LNA on
rxbuf_en	0	rw	**Enable RX buffer** 0_D: RX buffer off 1_D: RX buffer on

3.2.5 Register Reg1 – Direct access register

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

DAR_REG1_REG **Address:** $0x01_H$

Register assignment of Reg1 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res			**div_bias_en**	**Res**			**qs_rd_en**
rw			rw	rw			rw

7	6	5	4	3	2	1	0
bb_dig_det_en	**Res**	**bb_boost_dis**	**bb_clk_chop_en**	**Res**	**bb_strup_hp**	**bb_amp_en**	**bb_sample_en**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Res	15:13	rw	**Not Used. Do not change reset values.**
div_bias_en	12	rw	**Enable divider bias** 0_D: Divider bias off 1_D: Divider bias on
Res	11:9	rw	**Not Used. Do not change reset values.**
qs_rd_en	8	rw	**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	7	rw	**Enable digital baseband detector** Enables digital part of detector. After first switching on of this bit after startup/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
Res	6	rw	**Not Used. Do not change reset values.**
bb_boost_dis	5	rw	**Baseband sample and hold switch boost setting** 0_D: SandH gate voltage boost is enabled (pulsed mode) 1_D: SandH gate voltage boost is disabled (CW mode)
bb_clk_chop_en	4	rw	**Enable clock chop** This bit enables continues clock signal for chopping. 0_D: Clock off 1_D: Clock on
Res	3	rw	**Not Used. Do not change reset values.**

(table continues...)

(continued)

Field	Bits	Type	Description
bb_strup_hp	2	rw	**Baseband startup boost mode** 0_D: Startup boost mode disabled 1_D: Startup boost mode enabled
bb_amp_en	1	rw	**Enable baseband amplifier** 0_D: Baseband amplifier disabled 1_D: Baseband amplifier enabled
bb_sample_en	0	rw	**Enable baseband sampling** Controls connection of mixer output to sample and hold capacitance. 0_D: Disconnected, hold phase 1_D: Connected, sampling phase

3.2.6 Register Reg2 – Threshold

Value after init sequence: Depends on QS2

THOLD_REG2_REG **Address:** $0x02_H$

Register assignment of Reg2 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
hprt	**aprt**	**dir_mode**	**thrs**				
rw	rw	rw	rw				
7	6	5	4	3	2	1	0
thrs							
rw							

Field	Bits	Type	Description
hprt	15	rw	**High pulse repetition time** After init sequence this is the inverse of the level on SPICSN at boot time. 0_D: No change 1_D: PRT is constantly multiplied by prt_mult (Reg13[1:0]). This is for saving power, but may cause problems with direction detection.
aprt	14	rw	**Adaptive pulse repetition time** Default value: 1_D 0_D: No change 1_D: PRT is multiplied by prt_mult (Reg13[1:0]) as long as no target is detected. This is for saving power.
dir_mode	13	rw	**Direction detection mode** Default value: 1_D 0_D: Mode 1, PDet=0 and TDet=1, when no target is detected. 1_D: Mode 2, PDet=1 and TDet=1, when no target is detected.

(table continues...)

(continued)

Field	Bits	Type	Description
thrs	12:0	rw	**Detector threshold level** Default after init sequence (dep. 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

Value after init sequence: 0x0F3A
This register must not be changed when pll_en=1 (Reg0[8]).

PLL_CONFIG1_REG4_REG **Address:** $0x04_H$

Register assignment of Reg4 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
pll_dft_mux		**Res**	**pll_bias_dis**	**pll_lf_iso**	**pll_lf_r4_sel**	**pll_cl_loop_pmode**	**pll_lf_r2_sel**
rw		rw	rw	rw	rw	rw	rw

7	6	5	4	3	2	1	0
xosc_mode	**pll_fbdiv_cnt**	**pll_cp_icp_sel**			**pll_cp_mode**	**pll_pfd_rdt_sel**	
rw	rw	rw			rw	rw	

Field	Bits	Type	Description
pll_dft_mux	15:14	rw	**PLL data mux for DFT** Default after init sequence: 0_D The chip output SPIDO 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
Res	13	rw	**Not Used. Do not change reset values.**

(table continues...)

(continued)

Field	Bits	Type	Description
pll_bias_dis	12	rw	**Disable PLL bias** 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	11	rw	**Loopfilter isolation mode** Default after init sequence: 1_D 0_D: Isolation with charge-keeping buffer enabled 1_D: Isolation with switches only
pll_lf_r4_sel	10	rw	**Select loopfilter R4 setting** Default after init sequence: 1_D 0_D: 12.4 kΩ 1_D: 0.1 kΩ
pll_cl_loop_pmode	9	rw	**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	8	rw	**Select loopfilter R2 setting** Default after init sequence: 1_D 0_D: 21.6 kΩ 1_D: 18.7 kΩ
xosc_mode	7	rw	**XTAL oscillator mode** Default after init sequence: 0_D 0_D: Amplitude setting 1 1_D: Amplitude setting 2
pll_fbdiv_cnt	6	rw	**Feedback divider counter setting** Default after init sequence: 0_D 0_D: 60 GHz-mode cntA=21 dec for 38.4 MHz 1_D: 60 GHz-mode cntB=20 dec for 40 MHz

(table continues...)

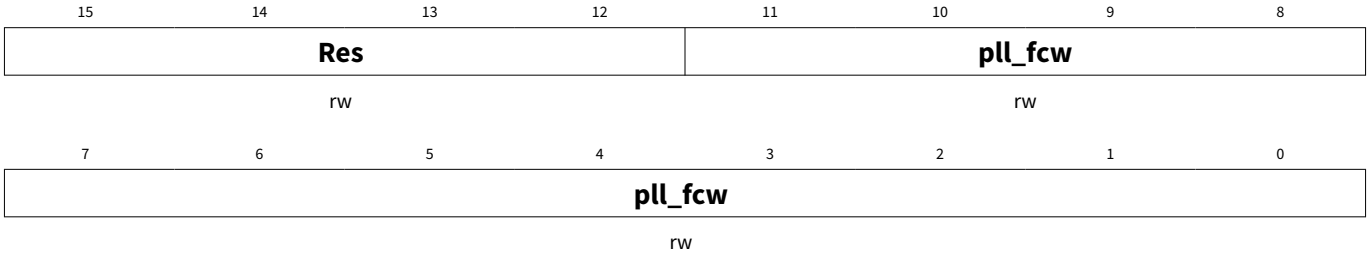
(continued)

Field	Bits	Type	Description
pll_cp_icp_sel	5:3	rw	**Charge pump current setting** Default after init sequence: 7_D 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	2	rw	**Charge pump bias mode** Default after init sequence: 0_D 0_D: Bias regulation loop active 1_D: Fix bias mode = bias regulation loop off
pll_pfd_rdt_sel	1:0	rw	**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

3.2.8 Register Reg5 – PLL config 2

Value after init sequence: Depends on QS4
This register must not be changed when pll_en=1 (Reg0[8]).

PLL_CONFIG2_REG5_REG **Address:** $0x05_H$
Register assignment of Reg5 **Reset value:** $0x0000_H$



Field	Bits	Type	Description
Res	15:12	rw	**Not Used. Do not change reset values.**

(table continues...)

(continued)

Field	Bits	Type	Description
pll_fcw	11:0	rw	**PLL frequency word** Default after init sequence (dep. on QS4) FCW for fstart (4-bit integer + 8-bit fractional) → 2.4 MHz raster @ 60 GHz Predefined settings for Japan mode (BGT60LTR11BAIP): 0xEA2: 60.6 GHz 0xECC: 60.7 GHz 0xEF5: 60.8 GHz 0xF1F: 60.9 GHz Predefined settings for Europe mode (BGT60LTR11AIP and BGT60LTR11SAIP): 0xF72: 61.1 GHz 0xF9C: 61.2 GHz 0xFC6: 61.3 GHz 0xFEF: 61.4 GHz

3.2.9 Register Reg6 – PLL config 3

Value after init sequence: 0x6800
This register must not be changed when pll_en=1 (Reg0[8]).

PLL_CONFIG3_REG6_REG **Address:** $0x06_H$
Register assignment of Reg6 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
pll_ld_tw_sel			**pll_ld_len**	**pll_ld_en**	**Res**		
rw			rw	rw	rw		

7	6	5	4	3	2	1	0
Res							
rw							

Field	Bits	Type	Description
pll_ld_tw_sel	15:13	rw	**Lock detection time window** Default after init sequence: 3_D 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	12	rw	**Lock detection - lock assertion condition + Lock detection - lock delay** Default after init sequence: 0_D 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	11	rw	**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
Res	10:0	rw	**Not Used. Do not change reset values.**

3.2.10 Register Reg7 – Duty cycling, timing, PD, MPA

Value after init sequence: 0x0457

DC_TMG_PD_MPA_REG7_REG **Address:** $0x07_H$

Register assignment of Reg7 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res				**dc_rep_rate**		**dc_on_pulse_len**	
rw				rw		rw	

7	6	5	4	3	2	1	0
Res	**vco2pll_dly**	**mpa2sh_dly**		**pd_en**	**mpa_ctrl**		
rw	rw	rw		rw	rw		

Field	Bits	Type	Description
Res	15:12	rw	**Not Used. Do not change reset values.**
dc_rep_rate	11:10	rw	**Duty cycle repetition rate** In Advance mode this is defined by inputs SPICLK and SPIDI. See Chapter 2.1.2. Default after init sequence: 1_D Defines the time until next pulsing sequence starts in pulsing mode. 0_D: 250 µs (17 km/h) 1_D: 500 µs 2_D: 1000 µs 3_D: 2000 µs
dc_on_pulse_len	9:8	rw	**Duty cycle on pulse length** Default after init sequence: 0_D 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
Res	7	rw	**Not Used. Do not change reset values.**
vco2pll_dly	6	rw	**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	5:4	rw	**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	3	rw	**Enable PD** Default after init sequence: 0_D 0_D: PD off 1_D: PD on

(table continues...)

(continued)

Field	Bits	Type	Description
mpa_ctrl	2:0	rw	**Medium power amplifier gain control** Default after init sequence: 7_D 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

Value after init sequence: 0x0000

DIV_REG8_REG **Address:** $0x08_H$

Register assignment of Reg8 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

7	6	5	4	3	2	1	0
Res				**div_sel**		**div_out_en**	**div_testmode_en**
rw				rw		rw	rw

Field	Bits	Type	Description
Res	15:4	rw	**Not Used. Do not change reset values.**
div_sel	3:2	rw	**Select divider** 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

(table continues...)

(continued)

Field	Bits	Type	Description
div_out_en	1	rw	**Enable divider out** Default: 0_D Enables the 2^14, 2^17, 2^21 divider logic. Setting div_out_en provides the divided output frequency from the VCO to Div_O pad based on div_sel value. The div_sel value of “0” is not applicable for this setting, if correspondent 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	0	rw	**Enable divider test mode** Default: 0_D Puts VCO frequency divided by 16 on pad Div_O. Overrides setting from bitfield div_sel. 0_D: Test mode off (div_sel active) 1_D: Test mode on

3.2.12 Register Reg9 – Baseband

Value after init sequence: 0x0068

BB_REG9_REG **Address:** $0x09_H$

Register assignment of Reg9 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

7	6	5	4	3	2	1	0
bb_hp_res		**bb_clk_chop_sel**	**bb_lpf_bw**	**bb_ctrl_gain**			
ro		ro	ro	rw			

Field	Bits	Type	Description
Res	15:8	rw	**Not Used. Do not change reset values.**
bb_hp_res	7:6	ro	**High pass filter resistor settings** Default: 1_D 0_D: 8 MΩ 1_D: 4 MΩ 2_D: 2 MΩ 3_D: 1 MΩ

(table continues...)

(continued)

Field	Bits	Type	Description
bb_clk_chop_sel	5	ro	**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	4	ro	**Low pass filter setting** Default: 0_D 0_D: 10 kHz 1_D: 60 kHz
bb_ctrl_gain	3:0	rw	**Baseband PGA gain setting** Default: 8_D 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 – Hold time

Value after init sequence: Depends on QS3

HT_REG10_REG **Address:** $0x0A_H$

Register assignment of Reg10 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
hold							
rw							

7	6	5	4	3	2	1	0
hold							
rw							

Field	Bits	Type	Description
hold	15:0	rw	**Hold time** Default after init sequence (dep. on QS3) Hold time for target detection in steps of 128 ms. However as the amplitude is filtered over 64 samples the shortest hold time is 16/32/64/128 ms depending on the PRT – and that minimum hold time is selected by “0”.

3.2.14 Register Reg12 – BITE

BITE_REG12_REG **Address:** $0x0C_H$

Register assignment of Reg12 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

7	6	5	4	3	2	1	0
bb_amux_ctrl		**bb_amux_en**	**bite_pd_en**	**bite_ctrl**			**bite_en**
rw		rw	rw	rw			rw

Field	Bits	Type	Description
Res	15:8	rw	**Not Used. Do not change reset values.**
bb_amux_ctrl	7:6	rw	**Select 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

(table continues...)

(continued)

Field	Bits	Type	Description
bb_amux_en	5	rw	**Enable analog voltage mux on QS4 pad** Default after init sequence: 0_D 0_D: AMUX off 1_D: AMUX on
bite_pd_en	4	rw	**Enable BITE power detector** Default after init sequence: 0_D 0_D: BITE PD off 1_D: BITE PD on
bite_ctrl	3:1	rw	**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	0	rw	**Enable BITE** Default after init sequence: 0_D 0_D: BITE disabled 1_D: BITE enabled

3.2.15 Register Reg13 – Algo 1

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

ALGO1_REG13_REG **Address:** $0x0D_H$

Register assignment of Reg13 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

7	6	5	4	3	2	1	0
phase_win_len			**mean_win_len**			**prt_mult**	
rw			rw			rw	

Field	Bits	Type	Description
Res	15:8	rw	**Not Used. Do not change reset values.**

(table continues...)

(continued)

Field	Bits	Type	Description
phase_win_len	7:5	rw	**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_D: 256 1_D: 512 2_D: 256 3_D: 128 4_D: 64 5_D: 256 6_D: 256 7_D: 256
mean_win_len	4:2	rw	**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_D: 256 1_D: 512 2_D: 256 3_D: 128 4_D: 64 5_D: 256 6_D: 256 7_D: 256
prt_mult	1:0	rw	**Pulse Repetition Time Multiplier** Default after init sequence: 0_D If the APRT is enabled by setting aprt (Reg2[14]) to "1", the multiplier is configurable by the following table: 0_D: 4 1_D: 8 2_D: 16 3_D: 2

3.2.16 Register Reg14 – Algo 2

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

ALGO2_REG14_REG **Address:** $0x0E_H$

Register assignment of Reg14 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
thrs_offset							
rw							

7	6	5	4	3	2	1	0
dir_hys_dis	**dir_keep**	**hold_x32**	**swap_iq**	**autoblind_dis**	**pulse_mon**	**phase_thrs**	
rw	rw	rw	rw	rw	rw	rw	

Field	Bits	Type	Description
thrs_offset	15:8	rw	**Threshold Offset** Default after init sequence: 0_D Possibility to shift the amplitude threshold up or down.
dir_hys_dis	7	rw	**Disable Direction Hysteresis** Default after init sequence: 0_D With this bit the direction detection can be tuned. 0_D: Some hysteresis is used to switch between directions 1_D: No hysteresis is used and the default is “departing”
dir_keep	6	rw	**Keep the Direction Algorithm Running** Default after init sequence: 0_D The behavior of the direction detection algorithm can be configured. 0_D: Only run while motion is detected. Otherwise output is “departing” 1_D: Keep the algorithm running during the hold time even if no motion is detected.
hold_x32	5	rw	**Multiply Hold Time by 32** Default after init sequence: 0_D 0_D: No change 1_D: Hold times are longer by a factor of 32
swap_iq	4	rw	**Swap I- and Q-Signal** Default after init sequence: 0_D 0_D: No change. 1_D: Swap IF signals for interpretation by direction algorithm. That leads to opposite direction detection.
autoblind_dis	3	rw	**Disable Blanking Off Sensor (“dead-time”)** Default after init sequence: 0_D Disable blanking algorithm after detection. Do not change!

(table continues...)

(continued)

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

3.2.17 Register Reg15 – Digital control

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

DIGI_CTRL_REG15_REG **Address:** $0x0F_H$

Register assignment of Reg15 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
soft_reset	**start_pm**	**clk_ext_dis**	**start_cw**	**fast_phase**	**dir_c2_1**		**fastmode**
rw	rw	rw	rw	rw	rw		rw

7	6	5	4	3	2	1	0
adc_mon	**miso_drv**	**mot_pol**	**dir_pol**	**stat_mux**			
rw	rw	rw	rw	rw			

Field	Bits	Type	Description
soft_reset	15	rw	**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_D: Reset inactive 1_D: Reset active

(table continues...)

(continued)

Field	Bits	Type	Description
start_pm	14	rw	**Start pulsed mode** Default after init sequence: 0_D With this bit it is possible to start the SPI pulsed mode (or CW mode). Typical use case 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_D: Inactive 1_D: Rising edge triggers pulsed mode (or CW mode if bit 12 is set)
clk_ext_dis	13	rw	**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 16-32 further clock edges are delivered. 0_D: Clock not disabled 1_D: Clock disabled
start_cw	12	rw	**Start CW mode instead** Default after init sequence: 0_D 0_D: No change 1_D: 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	11	rw	**Faster phase evaluation** Default after init sequence: 0_D 0_D: Start phase (direction) evaluation only when a target is detected. Therefore there is always some latency between TDet and PDet. 1_D: 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	10:9	rw	**Direction mode** Default after init sequence: 0_D Similar to bit “dir_mode” in Reg2. Do not change!
fastmode	8	rw	**SPI fast mode** Default after init sequence: 0_D 0_D: SPIDO changes on rising edge of SPICLK 1_D: SPIDO changes on falling edge of SPICLK
adc_mon	7	rw	**ADC monitoring** Default after init sequence: 0_D For in-depth-debugging only. Do not change!
miso_drv	6	rw	**SPI force MISO driver** Default after init sequence: 0_D 0_D: SPIDO is High-Z when SPICSN=1 1_D: SPIDO is always driven to some level

(table continues...)

(continued)

Field	Bits	Type	Description
mot_pol	5	rw	**Motion polarity** Default after init sequence: 0_D 0_D: TDet is low-active 1_D: TDet is high-active
dir_pol	4	rw	**Direction polarity** Default after init sequence: 0_D 0_D: PDet is low when departing 1_D: PDet is low when approaching
stat_mux	3:0	rw	**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 bits of Reg56 are replaced by the following: 1_D: 8-bit ADC reading of pad QS2 in Advance mode and "00" 2_D: 8-bit ADC reading of pad QS3 in Advance mode and "00" 3_D: 8-bit ADC reading of GND in Advance mode and "00" 4_D: 8-bit ADC reading of Vdd in Advance mode and "00" 5_D: "000000", advance_mode, SPICSN, SPIDI and SPICLK (pad states after reset) 6_D: Amplitude 7_D: Amplitude << 3 Others: qs1_s, init_done, qs2_s, qs3_s, qs4_s and advance_mode

3.2.18 Register Reg34 – ADC start

ADC_STRT_REG34_REG **Address:** $0x22_H$

Register assignment of Reg34 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

7	6	5	4	3	2	1	0
Res					**adc_en**	**bandgap_en**	**adc_clk_en**
rw					rw	rw	rw

Field	Bits	Type	Description
Res	15:3	rw	**Not Used. Do not change reset values.**
adc_en	2	rw	**Enable ADC block** Default: 0_D 0_D: ADC disabled 1_D: ADC enabled

(table continues...)

(continued)

Field	Bits	Type	Description
bandgap_en	1	rw	**Enable bandgap** Default: 0_D This bandgap is needed for ADC. 0_D: Bandgap disabled 1_D: Bandgap enabled
adc_clk_en	0	rw	**Enable local ADC clock** Default: 0_D 0_D: ADC clock disabled 1_D: ADC clock enabled

3.2.19 Register Reg35 – ADC convert

A write access to Reg35 starts ADC conversion with the selected settings, even if the same data is written into the register.

ADC_CNVT_REG35_REG **Address:** $0x23_H$

Register assignment of Reg35 **Reset value:** $0x0000_H$

15	14	13	12	11	10	9	8
Res							
rw							

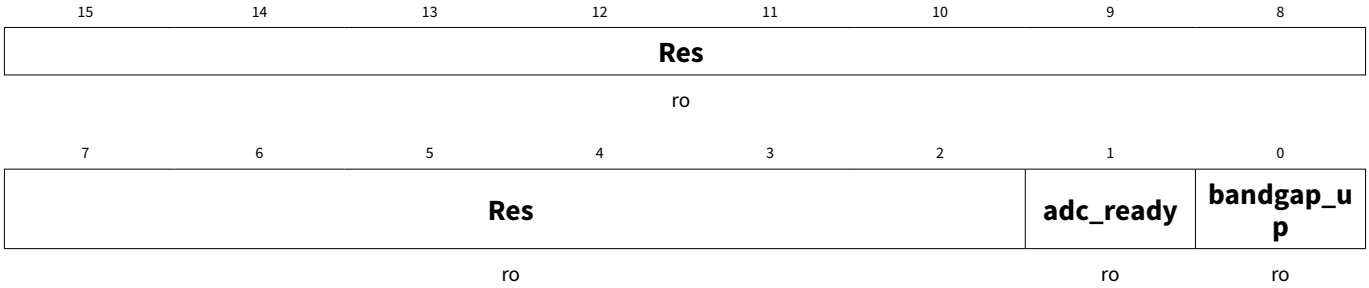
7	6	5	4	3	2	1	0
lv_gain	**Res**		**chnr_all**	**chnr**			
rw	rw		rw	rw			

Field	Bits	Type	Description
Res	15:8	rw	**Not Used. Do not change reset values.**
lv_gain	7	rw	**Analog input channel 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 1.613 V 1_D: gain = 1.00, fullscale analog input voltage 1.21 V
Res	6:5	rw	**Not Used. Do not change reset values.**
chnr_all	4	rw	**Channel number all** Default: 0_D 0_D: chnr selects channel to convert 1_D: Converts all 16 channels, chnr is ignored
chnr	3:0	rw	**Channel number** Default: 0_D Analog input channel number selected for sampling.

3.2.20 Register Reg36 – ADC status

ADC_STS_REG36_REG **Address:** $0x24_H$

Register assignment of Reg36 **Reset value:** $0x0000_H$



Field	Bits	Type	Description
Res	15:2	ro	**Not Used. Do not change reset values.**
adc_ready	1	ro	**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
bandgap_up	0	ro	**Bandgap up flag** 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

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 8-bit ADC also bit 0 and bit 1 are not used. Not used bits will deliver a zero when read.

Table 16 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 Reg56 – Status and chip version

Reset value: depending on chip_version and stat_mux (Reg15[3:0]) (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

STS_CHIP_VER_REG56_REG **Address:** $0x38_H$

Register assignment of Reg56 **Reset value:** H

15	14	13	12	11	10	9	8
qs1_s		**init_done**	**qs2_s**		**qs3_s**		**qs4_s**
ro		ro	ro		ro		ro

7	6	5	4	3	2	1	0
qs4_s	**advance_mode**	**Res**		**pll_lock_detect**	**chip_version**		
ro	ro	ro		ro	ro		

Field	Bits	Type	Description
qs1_s	15:14	ro	**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 = 100 kΩ to V_{DD} 11_B: QS1 = V_{DD}
init_done	13	ro	**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
qs2_s	12:11	ro	**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 = 100 kΩ to V_{DD} 11_B: QS2 = V_{DD}
qs3_s	10:9	ro	**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 = 100 kΩ to V_{DD} 11_B: QS3 = V_{DD}

(table continues...)

(continued)

Field	Bits	Type	Description
qs4_s	8:7	ro	**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 = 100 kΩ to V_{DD} 11_B: QS4 = V_{DD}
advance_mode	6	ro	**Advance Mode Indicator** Default after init sequence: 1_D This bit reflects the sampled PLL_Trig state. 0_D: Autonomous basic mode 1_D: Autonomous advance mode
Res	5:4	ro	**Not Used. Do not change reset values.**
pll_lock_detect	3	ro	**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	2:0	ro	**Chip version** Default: sample dependent - here 3 Every variant has its own version number, it is hard wired on analog top level. These bits are read only.

3.2.23 Register GSR0 – SPI status register

The global status register GSR0 is sent on SPIDO at the same time as the address and the read/write bit is sent on SPIDI, MSB leading. There is only one bit used, it is bit adc_result_ready (GSR0[2]). This is a flag for completed conversion. It is cleared by reading out any result register (Reg38-Reg53), adc_clk_en (Reg34[0]) has to be set to "1" for that.

SPI_STS_GSR0_REG **Address:** $_H$

Register assignment of GSR0 **Reset value:** $0x0000_H$

7	6	5	4	3	2	1	0
Res					**adc_result_ready**	**Res**	
ro					ro	ro	

Field	Bits	Type	Description
Res	7:3	ro	**Not Used. Do not change reset values.**
adc_result_ready	2	ro	**ADC result ready flag** Default: 0_D This flag indicates if the ADC conversion is completed, and the result is ready. 0_D: ADC result not ready. 1_D: ADC result ready.
Res	1:0	ro	**Not Used. Do not change reset values.**

4 Analog to digital converter

4.1 ADC conversion sequence

An ADC conversion consists of four different phases detailed below:

4.1.1 Enable bandgap

The bandgap is enabled by setting the bandgap_en bit (Reg34[1]). This can be done simultaneously with adc_clk_en (Reg34[0]). The bandgap can be enabled or disabled independently of all other parts of the ADC.

The startup time of the bandgap is temperature and MMIC device dependent. Enabling of the ADC is not allowed before the bandgap_up flag (Reg36[0]) is readout as high.

4.1.2 Enable local ADC clock

The local clock generator is enabled by setting the adc_clk_en bit (Reg34[0]), without setting any other bits, except bandgap_en.

4.1.3 Enable ADC

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

The adc_en bit (Reg34[2]) enables the ADC. The adc_ready flag (Reg36[1]) 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 Reg35, 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 (Reg38-Reg53).

4.2 ADC configuration

4.2.1 Analog input channel gain

By setting bit lv_gain (Reg35[7]) the gain for the analog input channels can be selected as follows:

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

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. Sampling time is fixed to 16 clock cycles.

4.2.3 ADC phases

The physical resolution is 8-bit.

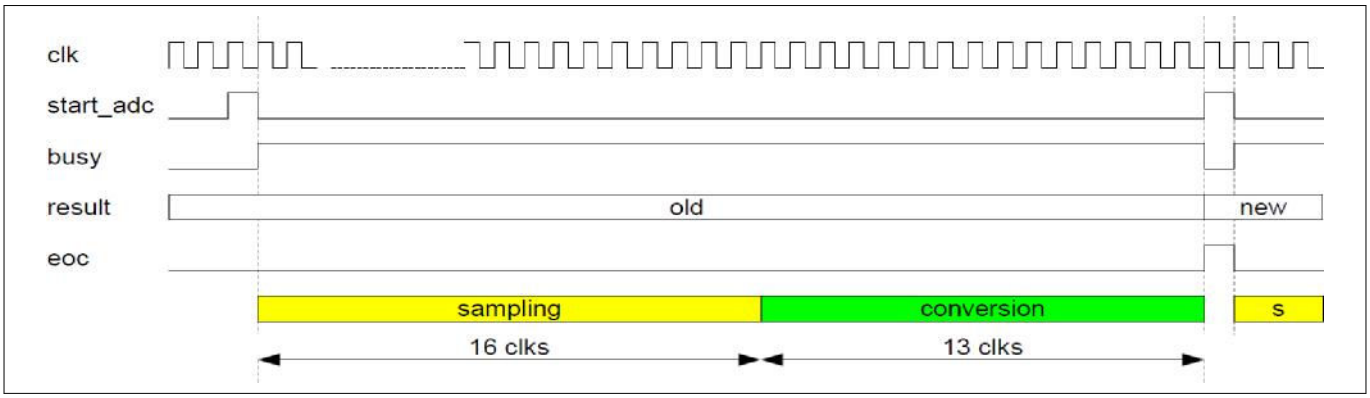


Figure 14 Timing diagram

4.3 Conversion time

An 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 15 MHz, max 50 MHz).

$$t_{conv} = \left(t_{sample} + t_{distrib} + t_{epcal}\right) * t_{adc_clk} = \left(16 + 13 + 12\right) * t_{adc_clk} \quad (1)$$

$$t_{conv_min} = \left(t_{sample} + t_{distrib} + t_{epcal}\right) * t_{adc_clk_50M} = \left(16 + 13 + 12\right) * \left(1/50e6\right) = 0.82\ \mu s \quad (2)$$

$$t_{conv_max} = \left(t_{sample} + t_{distrib} + t_{epcal}\right) * t_{adc_clk_15M} = \left(16 + 13 + 12\right) * \left(1/15e6\right) = 2.73\ \mu s \quad (3)$$

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_en 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_en 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 sensor ADC and for setting of TDet/PDet outputs of the BGT60LTR11AIP.

Target detected (TDet) output is used to show if a motion is detected or not. TDet is an active low pin. Thus, it's set low when a motion is detected, otherwise high.

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 50 ms after setting of the bb_dig_det_en (Reg1[7]) to allow the settling of baseband circuit.

5.2 Hold time

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

It can be configured in Reg10.

Revision history

Document revision	Date	Description of changes
1.00	2020-09-09	First preliminary release
1.10	2020-10-06	Added autonomous mode
1.20	2021-07-15	Changes all over the document
1.30	2021-10-11	Changes all over the document
1.40	2021-11-17	Support BGT60LTR11BAIP Japanese MMIC version
1.50	2022-08-01	Support BGT60LTR11SAIP down-specified MMIC version
1.60	2023-02-14	Miscellaneous document cleanup updates
1.70	2023-11-14	Changed Reg9 bit fields bb_hp_res, bb_clk_chop_sel and bb_lpf_bw from read-write to read-only
1.80	2024-07-09	Editorial changes

Trademarks

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

Edition 2024-07-09

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2024 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-ahh1628527864346

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.

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.