User guide to BGT60CUTR13AIP

XENSIV™ 60 GHz CMOS radar sensors

About this document

Scope and purpose

This document describes the functionality of XENSIV™ BGT60CUTR13AIP next-generation, ultra-low-power 60 GHz CMOS radar sensor with integrated processing for IoT applications and describes each module functionality in detail. The document covers:

- Clock and reset
- Device communication via SPI
- RF front-end setup
- Configuration of the frames
- Data processing

Additionally, the document provides guidance on configuring certain modules and offers programming guidelines. It also includes RF shield board design guidelines, along with simulation and measurement results.

Intended Audience

The intended audiences for this document are engineers interested in or working with XENSIV™ BGT60CUTR13AIP device.

About this product group

Target applications

- Security camera and video doorbell
- Heating ventilation and air conditioning (HVAC)
- Smart home
- Smart TVs
- Smart thermostat
- Consumer electronics
- Robotics

Product family

Infineon's new XENSIV™ 60 GHz CMOS mmWave radar sensors offer advanced, high-precision sensing with ultra-low power consumption. The integrated hardware accelerator and the bootloader enable autonomous operation and offload complex tasks from the host MCU by running pre-processing or full motion detection on chip. With improved bandwidth and output power, the CMOS product family achieves high resolution and detection range, supporting applications from motion detection and presence sensing to people tracking, gesture control, and vital-sign sensing.

Table of contents

About this document ... **1**
About this product group ... **1**
Table of contents ... **2**
1 Introduction ... **6**
2 Pin configuration ... **7**
2.1 Pin assignment ... 7
2.2 Pin definition and functions ... 8
3 Interfaces ... **10**
3.1 System clock ... 10
3.1.1 Embedded XTAL oscillator ... 10
3.1.2 External oscillator ... 10
3.1.3 Oscillator output ... 11
3.2 Serial Peripheral Interface (SPI) ... 11
3.2.1 Advanced features ... 12
3.2.2 Standard timing ... 12
3.2.3 High-speed timing ... 13
3.2.4 Operation modes ... 13
3.2.5 SPI error handling ... 16
3.2.5.1 SPI clock number error ... 16
3.2.5.2 SPI transaction (word) number error ... 17
3.2.5.3 SPI no system clock error ... 17
3.3 Digital input output pins ... 18
3.3.1 Configuration ... 18
3.3.2 Mapping ... 18
3.3.3 Pulse extension ... 21
3.3.4 Trigger ... 22
4 Device initialization ... **24**
4.1 Power supply startup ... 25
4.2 Reset and clock startup ... 25
4.2.1 Hardware and power-on reset (POR) ... 25
4.2.2 Bootloader (BL) ... 26
4.2.2.1 Functionality ... 26
4.2.3 RC oscillator (RCOSC) ... 27
4.2.4 XTAL oscillator (XOSC) ... 28
4.2.4.1 Manual software startup ... 28
4.2.4.2 Automatic hardware startup ... 29
4.2.5 External oscillator (EXT_OSC) ... 29
4.2.5.1 Manual software startup (external clock) ... 29
4.2.5.2 Automatic hardware startup (external clock) ... 30
4.2.6 Frequency doubler ... 30
4.3 Trimming and calibration ... 32
4.3.1 RC oscillator trimming ... 32
4.3.1.1 Automatic hardware trimming ... 32
4.3.1.2 Manual software trimming ... 32
4.3.2 ADC calibration ... 33
4.3.3 FD duty cycle trimming ... 34
4.4 Configuration ... 34

4.4.1 Frame configuration 34
4.4.2 Clock configuration 35
4.5 Quick start 35
5 Function block description 37
5.1 Identification (ID) 37
5.2 General timer (GTMR) 37
5.3 Wakeup controller (WUC) 38
5.4 Power domain controller (PDC) 39
5.5 Interrupt router (IR) 40
5.6 Phase lock loop (PLL) 42
5.6.1 PLL parameter calculation 43
5.6.2 PLL timings 44
5.6.3 PLL configuration 45
5.7 Radio frequency front-end (RFFE) 45
5.7.1 Transmitter (TX) 46
5.7.2 Receiver (RX) 47
5.7.3 Analog baseband (ABB) 48
5.7.3.1 High-pass filter (HPF) 48
5.7.3.2 Variable gain amplifier (VGA) 49
5.7.3.3 Anti-aliasing filter (AAF) 50
5.7.4 Analog digital converter (ADC) 50
5.7.5 Phase shifter (PHS) 51
5.8 Sensor ADC (SADC) 52
5.8.1 SADC result threshold 53
5.9 Linear feedback shift register (LFSR) 53
5.9.1 Binary phase-shift keying (BPSK) pattern generation 54
5.9.2 Test data generation 55
6 Functional description 56
6.1 Radar frame definition 56
6.1.1 Sequencer (SEQ) 57
6.1.1.1 Overview 58
6.1.1.2 Commands 58
6.1.1.3 Memory 60
6.1.1.4 Error handling 61
6.1.1.5 Debugging and testing 62
6.1.2 Chirp timing 62
6.1.3 Extended bandwidth (interlaced) chirps 63
6.2 Data flow 64
6.2.1 Oversampling and decimation 64
6.2.2 Chirp averaging (AVG) 65
6.2.3 First-in first-out (FIFO) – Data memory 66
6.2.3.1 Data structure 67
6.2.3.2 Data header 68
7 Presence detection (PD) 70
7.1.1 Overview 70
7.1.2 Operation modes 70
7.1.2.1 Ultra-low power (ULP) 70
7.1.2.2 Range Doppler (RD) 74
7.1.3 Algorithm selection 76

7.1.3.1 Moving target indicator (MTI) 77
7.1.3.2 Windowing 77
7.1.3.3 Range – Real FFT (RFFT) 79
7.1.3.4 Doppler – Complex FFT (CFFT) 81
7.1.3.5 Coordinate rotation digital computer (CORDIC) 82
7.1.3.6 Constant false alarm rate (CFAR) 82
7.1.3.7 Merge 87
7.1.3.8 Local maximum 87
7.1.3.9 Target extraction 88
7.1.3.10 Masking 92
7.1.3.11 Event counter 98
7.1.4 Data access (RX_BUFFER) 99
7.1.5 Data forwarding (FIFO) 101
7.1.5.1 Timing 102
7.1.5.2 Data layout 103
7.1.6 Clutter handling 110
8 ROM configuration 111
8.1 Config 0 111
8.1.1 Field of view experiment for maximum detectable range validation 113
8.1.2 Operating modes 114
8.1.2.1 Hardware reset/power cycle (no SPI required) 114
8.1.2.2 Interrupt cleared via SPI command 115
8.1.2.3 Motion-triggered power control via SPI 115
8.1.2.4 Interrupt-based power cycling 116
8.2 Config 1 116
8.2.1 Outdoor motion detection experiment and performance validation 119
8.2.2 Operating modes 120
8.2.2.1 Hardware reset/power cycle (no SPI required) 120
8.2.2.2 Motion-triggered power control (no SPI required) 121
8.2.2.3 Interrupt Handling with High Frame Rate Switching (SPI Required) 121
8.3 Config 2 122
8.3.1 Operating modes 125
8.3.1.1 Default operating mode (auto interrupt clear) 125
8.3.1.2 Hardware reset/power cycle (no SPI required) 125
8.3.1.3 Motion-triggered power control (No SPI required) 126
8.3.1.4 Interrupt clearing via SPI command (SPI required) 126
9 PCB design guideline 127
9.1 XENSIV™ BGT60CUTR13AIP PCB schematics design guide 127
9.2 XENSIV™ BGT60CUTR13AIP pin assignment for minimum configuration 127
9.2.1 XENSIV™ BGT60CUTR13AIP schematic for minimal configuration 128
9.2.2 XENSIV™ BGT60CUTR13AIP supply domains and bypass capacitors 128
9.3 External clocks 130
9.3.1 External 80 MHz XTAL: 131
9.3.2 External 40 MHz XTAL: 131
9.3.3 XTAL Remarks 131
9.4 External MMIC power supply 132
9.4.1 Supply noise requirements: maximum spurious signal levels 132
9.4.2 LDO schematics 134
9.4.3 LDOs (low-dropout regulators) and DC-DC converters 134

9.4.4 Supply power-up sequence 135
9.4.5 Supply filtering 136
10 RF shield design, simulation, and measurement 137
10.1 Radar mode simulation setup 137
10.1.1 Far field computation 137
10.1.2 Radar matrix computation 138
10.1.3 Data post-processing 141
10.1.3.1 Algorithm flow diagram for field of view (FoV) angle plots 142
10.2 Radar shield PCB setup 144
10.3 Small PCB – Largest side ≤ $5\lambda_0$ (≈25 mm) 145
10.3.1 Radar performance in E-plane 146
10.3.1.1 Step 1 – Choose the right PCB stack up 147
10.3.1.2 Step 2 – Optimize the layout of the top metal layer 147
10.3.2 H-plane performance – Small PCB 149
10.4 Large PCB – Largest side > $5\lambda_0$ (≈25 mm) 150
10.4.1 E-plane performance – Large PCB 151
10.4.2 H-plane performance – Large PCB 152
10.5 Impact of radome on radar performance 152
10.5.1 Introduction 152
10.5.2 SBR+ simulation setup 152
10.5.3 Recommended radome thickness 154
10.5.4 Recommended airgap 155
10.5.5 Conclusion 155
10.6 Design guidelines of CAD and PCB hardware integration 158
10.7 Measurement setup 160
10.7.1 Introduction 160
10.7.2 Measurement environments 160
10.7.3 Tx characterization 161
10.7.3.1 Measurement results 163
10.7.4 Rx characterization 163
10.7.4.1 Measurement results 164
10.7.5 Radar mode measurement setup 166
10.7.5.1 Measurement results 167
11 Related resources 169
References 170
Glossary 171
Revision history 175
Disclaimer 177

1 Introduction

The XENSIV™ BGT60CUTR13AIP sensor is a radar system-on-chip (SoC). It can operate in frequency-modulated continuous wave (FMCW) mode and for test purposes in continuous wave (CW) mode. The SoC consists of several analog and digital blocks: radio frequency (RF) front-end, analog baseband (ABB), analog-to-digital converter (ADC), phase-locked loop (PLL), FIFO (first-in first-out) memory, Serial Peripheral Interface (SPI) and antennas. For power efficiency, XENSIV™ BGT60CUTR13AIP SoC features an embedded hardware accelerator (HWA)-based preprocessing which helps reduce data transfer to the host controller.

The device is optimized for high power efficiency by integrating domains specific low drop-out voltage regulators (LDO), separating logic into power domains which are dynamically controlled by hardware and integrating both low-power RC oscillator (RCOSC) as well as high precision crystal oscillator (XOSC) on-chip.

To offload the microcontroller or application processor, the sensor features a fully embedded stand-alone motion/presence detection (MD) processing module that notifies the host via an interrupt once a target is moving within its field of view (FoV). The block diagram of XENSIV™ BGT60CUTR13AIP is shown in Figure 1.

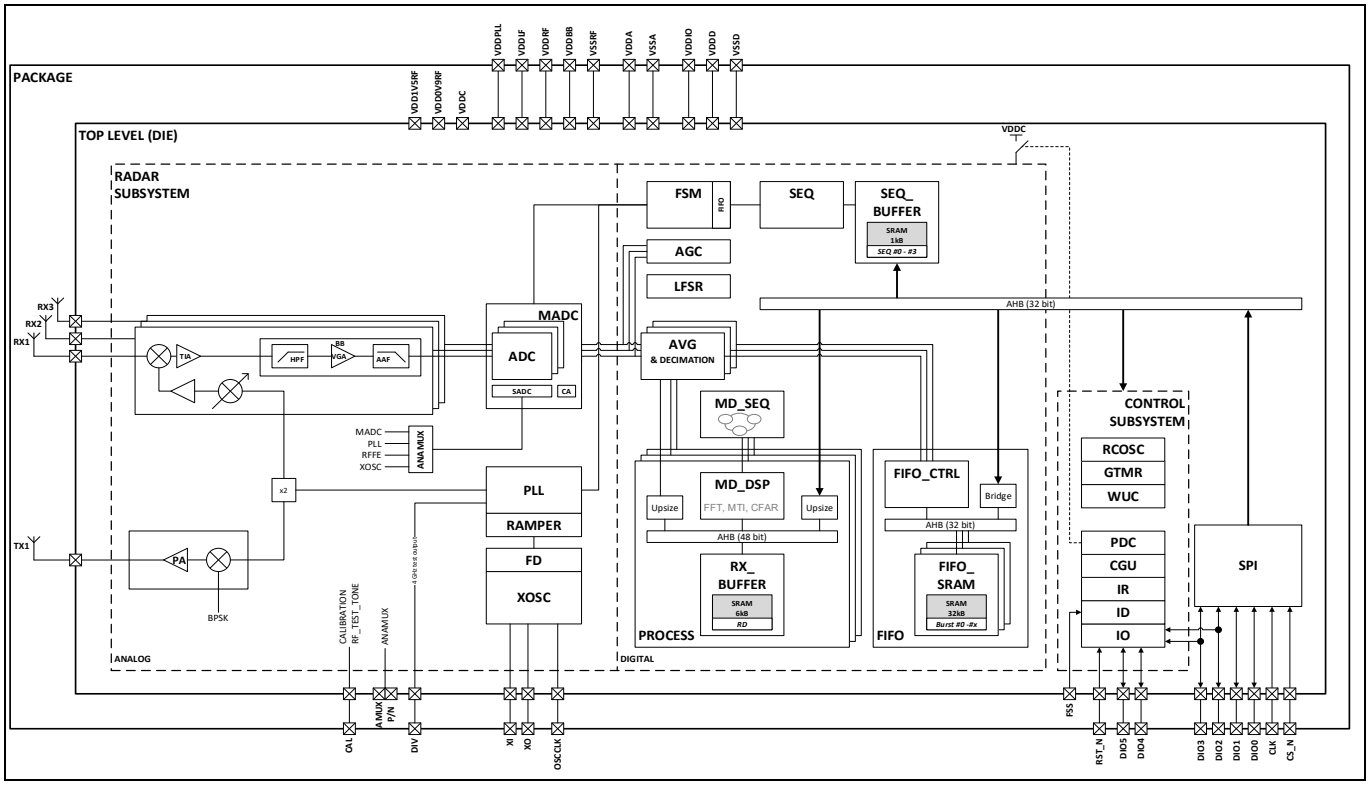


Figure 1 Block diagram of XENSIV™ BGT60CUTR13AIP radar sensor

2 Pin configuration

Figure 2 shows the transparent top view of the laminate package with the pin and antenna number assignment, as shown in Table 1.

2.1 Pin assignment

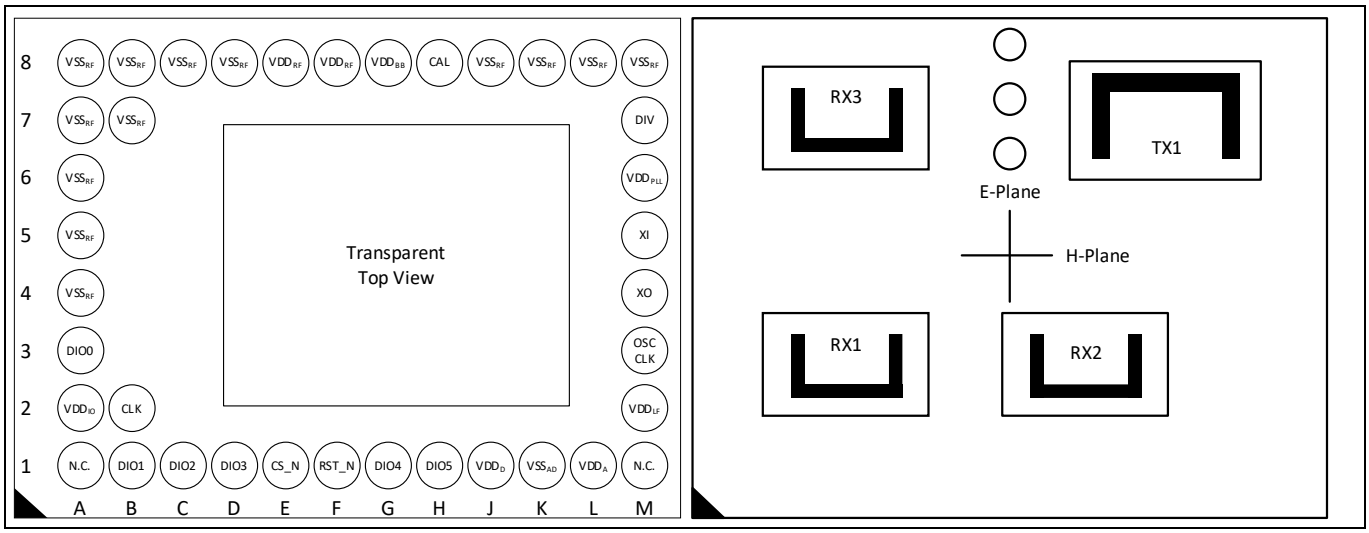


Figure 2 XENSIV™ BGT60CUTR13AIP pin assignment

Table 1 XENSIV™ BGT60CUTR13AIP radar sensor pin assignment

Ball	Symbol
A1	Not connected (NC)
A2	VDDIO
A3	DIO0
A4	VSSRF
A5	VSSRF
A6	VSSRF
A7	VSSRF
A8	VSSRF
B1	DIO1
B2	CLK
B7	VSSRF
B8	VSSRF
C1	DIO2
C8	VSSRF
D1	DIO3
D8	VSSRF
E1	CS_N
E8	VDDRF
F1	RST_N

Ball	Symbol
F8	VDDRF
G1	DIO4
G8	VDDBB
H1	DIO5
H8	CAL
J1	VDDD
J8	VSSRF
K1	VSSAD
K8	VSSRF
L1	VDDA
L8	VSSRF
M1	Not connected
M2	VDDLF
M3	OSCCLK
M4	XO
M5	XI
M6	VDDPLL
M7	DIV
M8	VSSRF

2.2 Pin definition and functions

Table 2 and Figure 3 show the pin definition and function mapping of XENSIV™ BGT60CUTR13AIP radar sensor.

Table 2 Pin definition and functions

Symbol	Direction	Type	Domain	Description
Supply pins				
VDDD	Input	Power	Supply	Supply for digital LDO
VDDPLL	Input	Power	Supply	Supply for PLL
VDDLF	Input	Power	Supply	Supply for PLL loop filter
VDDA	Input	Power	Supply	Supply for ADC
VDDBB	Input	Power	Supply	Supply for baseband
VDDRF	Input	Power	Supply	Supply for RF
VDDIO	Input	Power	Supply	Supply for I/O pad
VSSRF	Input	Ground	Supply	Ground for PLL and RF
VSSAD	Input	Ground	Supply	Ground for ADC and digital
Test pins				
DIV	Output	Analog	RF	Divided (/16) VCO output
CAL	Input	Analog	RF	PD calibration
Input output pins				

Symbol	Direction	Type	Domain	Description
XI	Input	Analog	LF	XOSC input (XTAL) or external clock input
XO	Output	Analog	LF	XOSC output (XTAL)
OSCCLK	Output	Analog	LF	XOSC output clock
RST_N	Input	Digital	I/O	Chip reset (active low)
CS_N	Input	Digital	I/O	SPI chip select (active low)
CLK	Input	Digital	I/O	SPI clock
DIO0	Input Output	Digital	I/O	SPI MOSI SPI I/O0
DIO1	Input Output	Digital	I/O	SPI MISO SPI I/O1
DIO2	Input Output	Digital	I/O	DIO SPI I/O2
DIO3	Input Output	Digital	I/O	DIO SPI I/O3
DIO4	Input Output	Digital	I/O	DIO
DIO5	Input Output	Digital	I/O	DIO

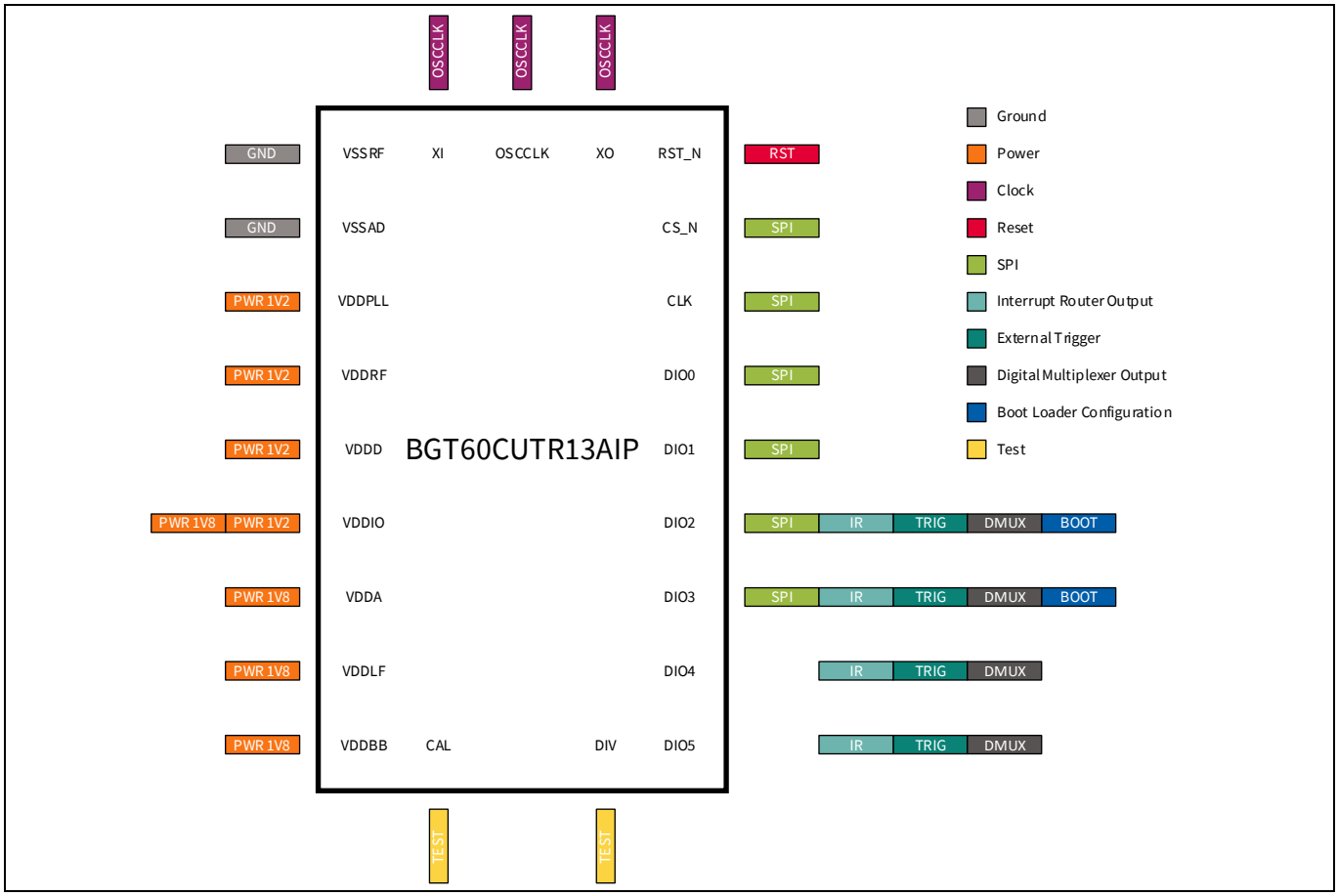


Figure 3 Pin function mapping

3 Interfaces

3.1 System clock

The device operates with an 80 MHz system clock. This clock can be provided either by an external oscillator via the XI pin or generated internally by the MMIC itself using the embedded crystal oscillator. The latter one requires an external crystal and capacitors.

3.1.1 Embedded XTAL oscillator

One option to generate the required 80 MHz system clock is to use the embedded crystal oscillator. For this a crystal resonator and two additional capacitors need to be connected to the XI and XO pins of the device as shown in Figure 4. The embedded crystal oscillator will work with either a 40 MHz or 80 MHz crystal resonator.

Note: When the 40 MHz crystal resonator is used, the internal frequency doubler must be enabled.

Depending on the application, the XOSC can be controlled either by software via SPI or by hardware via the embedded wakeup controller (WUC) and must be configured via `CLK_XOSC_CTRL0.MODE`. In case the user wants to have full control over the XOSC, its start-up routine needs to be executed by the host, ensuring the proper timings (`MODE=1`$_D$). There is also the option of giving control over starting up and powering down the XOSC to the device itself (`MODE=2`$_D$). In this mode, the internal wakeup controller (WUC) manages all timings during the startup routine.

Note: When MODE=2 is selected, ensure that the RCOSC is enabled, as the WUC requires the RCOSC_CLK.

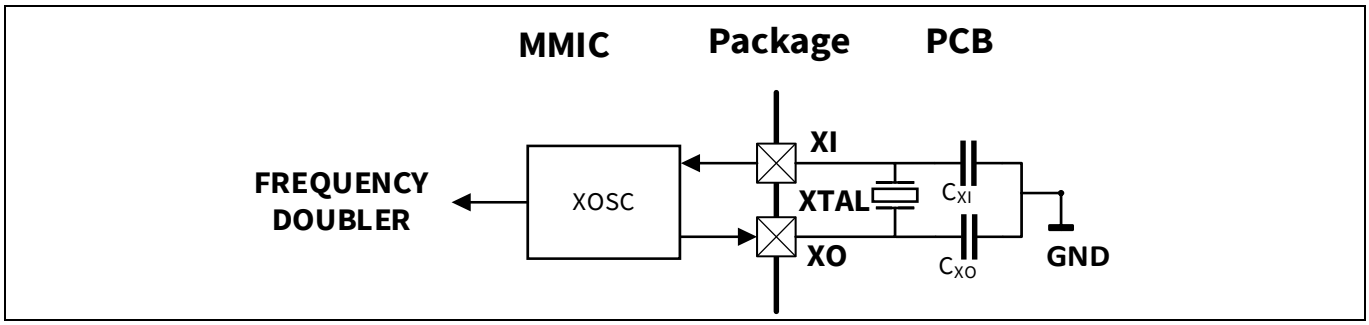


Figure 4 Embedded crystal oscillator

3.1.2 External oscillator

In addition to the embedded oscillator, another option to get the required system clock is to use an external active oscillator or to use an available clock in the system. In such cases, the clock can be fed in via the XI pin as shown in Figure 5 and needs to be configured with `CLK_XOSC_CTRL0.MODE`.

As the XOSC core itself is not required, in such a case, there is a slightly different startup routine. Again, there is the option to either control the clock generation (shaper) by hardware (`MODE=3`$_D$) via the integrated wakeup controller (WUC) module or by software (`MODE=4`$_D$).

Note: When MODE=3 is selected, ensure that the RCOSC is enabled as the WUC requires the RCOSC_CLK.

Note: When MODE=4 needs to be selected, ensure that the state transition is done via the OFF (MODE=0) state.

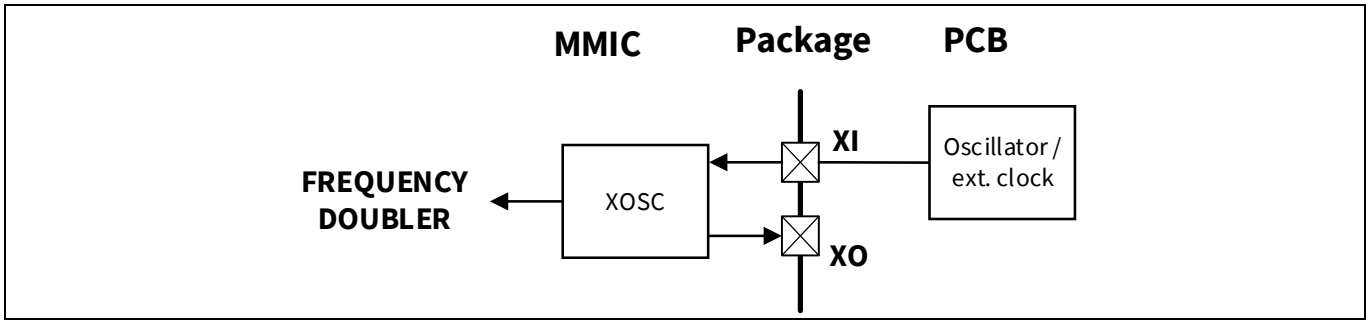


Figure 5 External oscillator clock

3.1.3 Oscillator output

The device offers the capability to provide the internally generated clock on the device interface via the OSCCLK pin, as shown in Figure 6. This feature can be used to clock an external microcontroller or other MMIC devices in setups involving multiple radar sensors. It must be enabled via `CLK_XOSC_CTRL1.CLK_OUT_EN`.

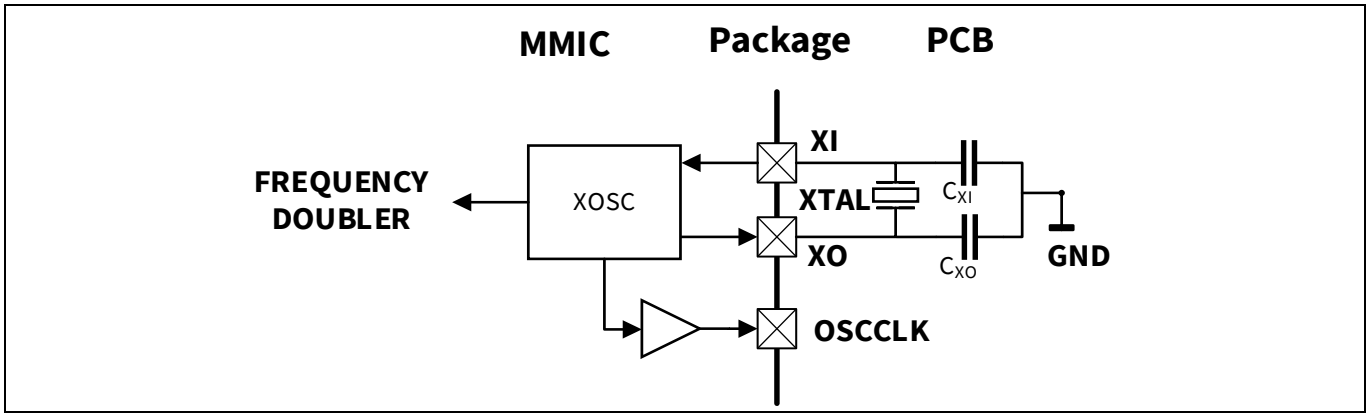


Figure 6 XTAL oscillator clock output

3.2 Serial Peripheral Interface (SPI)

For communication with the device, a Serial Peripheral Interface (SPI) with mode 0 is provided. It is used to configure the chip to operate in the targeted mode and read out the captured sensor data.

The device features up to six DIO pins for SPI communication, whereas four of them are static in terms of functionality and two of them are programmable based on the application requirements.

- CS_N to be connected to SS of the SPI master
- CLK to be connected to the CLK of the SPI master
- DIO0 to be connected to the MOSI/IO0 of the SPI master
- DIO1 to be connected to the MISO/IO1 of the SPI master
- DIO2 to be connected to the IO2 of the SPI master (optional)
- DIO3 to be connected to the IO3 of the SPI master (optional)

The device operates in mode 0 with clock polarity (CPOL) and clock phase (CPH) zero.

3.2.1 Advanced features

Compared to the standard SPI protocol, the sensor features three proprietary functions which may be useful in certain types of applications.

- SPI broadcast to configure multiple devices in parallel
- Option to change LSB first and MSB first
- Flexibility with high SPI clock speeds

The first feature is called “SPI broadcast” and can be used to program several identical XENSIV™ BGT60CUTR13AIP devices in parallel to reduce configuration time. It can also be used to trigger a frame start operation for multiple devices by software via SPI. In such cases, multiple devices start frame execution at the same time. To operate in a time-division multiplexed (TDM) scenario, the timing offset at the very beginning of each frame needs to be different to avoid an overlap in the time domain. SPI is a shared bus communication interface which means sharing the same physical lanes for multiple slaves and the output driver of the sensor for the MISO lane needs to be disabled to avoid any data collision. Therefore, as long as the `SPI_CTRL.BC` bit is set, the output drivers are disabled. Once the broadcast operation is over, the control bit needs to be cleared by the host.

The second non-standard feature provides the option to change between “LSB first and MSB first”. When `SPI_CTRL.MSB_SEL` is set, NUM, ADDRESS, and DATA fields in the SPI protocol are interpreted with MSB first; otherwise, with LSB first. By default, this bit field is cleared, so after a hardware or software reset, the SPI protocol is handled with LSB first.

The third feature could be helpful for systems with “high SPI clock” speed. The quasi-standard SPI protocol defines sending out the data on slave side with the falling edge of the SPI clock and registering the data on the master side with the rising edge of the SPI clock. However, based on the load and the length of the MISO line, as well as the output driver strength and the setup time required by the SPI master, it could be challenging to fulfill the timing requirements for high SPI clock speeds. Therefore, the device offers the capability to send out the data half a clock cycle earlier when `SPI_CTRL.EDGE` is set to `RISING(=1B)`. The timing details are shown in Section “High-speed timing”.

3.2.2 Standard timing

The timing diagram for standard SPI timing is shown in Figure 7. An SPI transfer begins with a falling edge of chip select signal CS_N generated by the SPI master. At the same time, the SPI master drives the level of the data input signal DIO0 (MOSI) according to the first bit to be transmitted. Also, with the falling edge of the chip select signal CS_N, the SPI slave applies the level of the data on the output signal DIO1 (MISO) according to the first bit to be transferred to the SPI master. The level becomes stable after the period t_{ds}. The SPI master must wait for the time t_L before the clock signal CLK can be generated.

With the rising edge of CLK, the SPI slave captures the level of DIO0. The SPI master must keep the DIO0 level stable for t_{sis} before and for t_{sih} after the rising edge of CLK to ensure valid setup and hold time of the SPI slave. With the falling edge of CLK, the SPI master shall set the level of DIO0 according to the next bit the master wants to send. The SPI master is supposed to read the level of DIO1 with the rising edge of CLK. The SPI slave keeps the DIO1 level stable for t_{soh} after the falling edge of CLK.

With the falling edge of CLK, the SPI slave drives the level of DIO1 according to the next bit. DIO1 becomes stable after the latest t_{sov}. After the last bit has been transferred and CLK has gone to low level, the SPI master must set CSN to high level to stop the transfer. The master must ensure that the period between the last rising edge of CLK and the rising edge of CS_N is not shorter than t_T. Within the period t_{dh} after the rising edge of CS_N, the SPI slave drives DIO1 to high impedance state again.

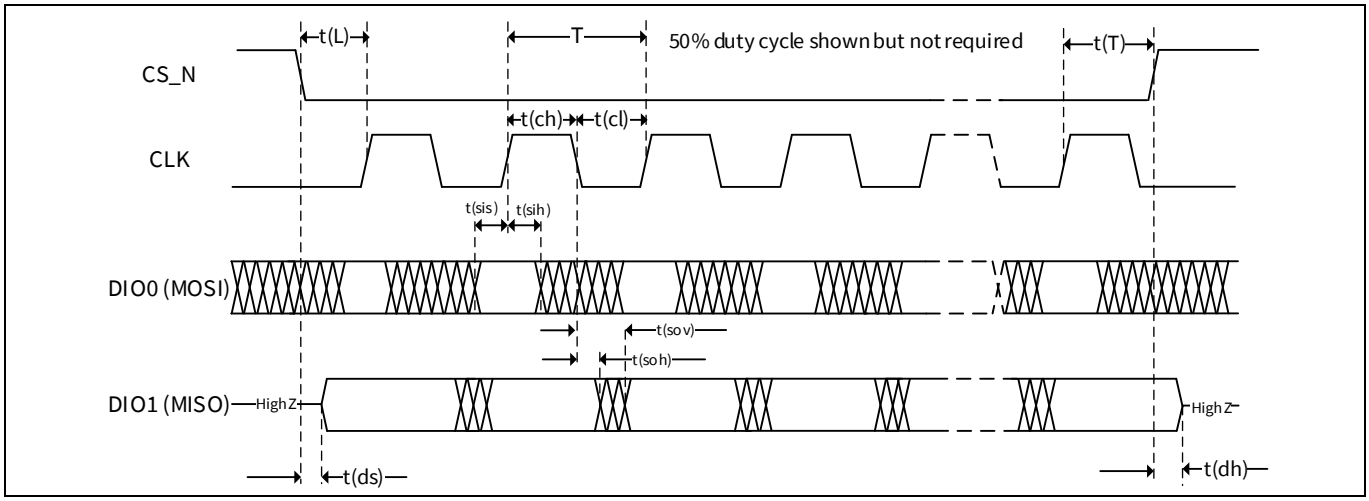


Figure 7 SPI standard timing

3.2.3 High-speed timing

The device can operate at SPI clock speeds up to 50 MHz, but the maximum achievable SPI clock speed is limited by input related setup and hold times of SPI master and SPI slave. In case the SPI master requires a longer setup time than T/2 - t_{sov}, the SPI clock speed in the standard SPI must be reduced. However, this device offers a mode that sends out the data earlier to still allow higher speeds. The timing diagram for this SPI mode is shown in Figure 8. The SPI master is still supposed to capture the level of DIO1 with the rising edge of CLK. The SPI slave keeps the level of DIO1 stable for t_{soh} after the rising edge of CLK and then sets the level of DIO1 according to the next bit which is sent out. This alternative SPI mode enables frequencies higher than 25 MHz.

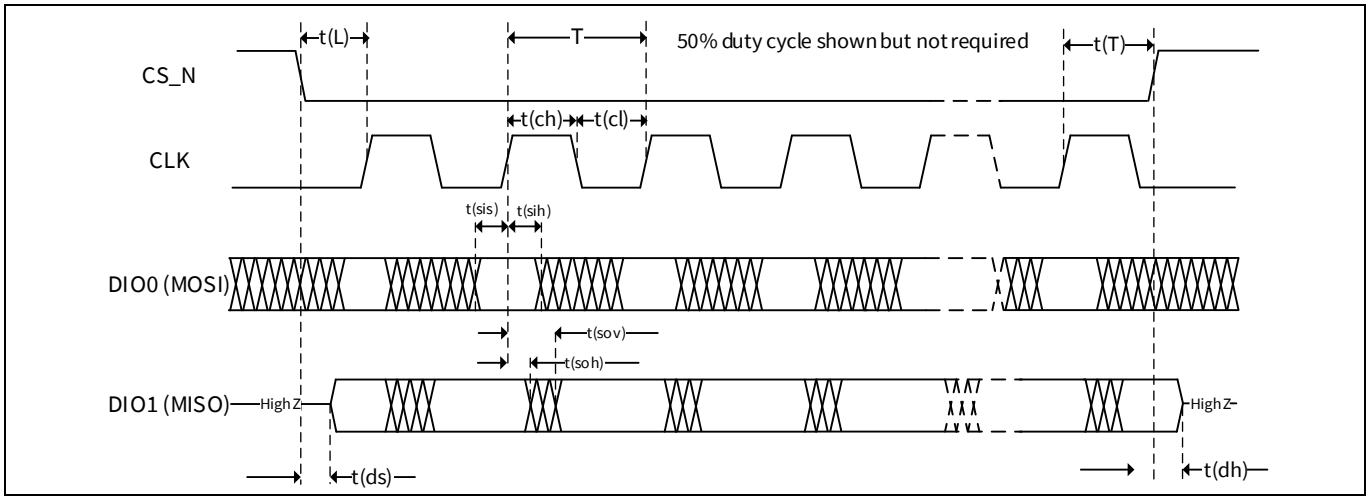


Figure 8 SPI high-speed timing

3.2.4 Operation modes

Each SPI transaction consists of three phases:

- Command phase
- Address phase
- Data phase

The command phase begins with a one-bit field called “WR (read:0B, write: 1B)”, which specifies the type of operation, either read or write. This is followed by a seven-bit field called “NUM”, which indicates the total number of intended data words per transaction. As the minimum number of 32-bit data words is 1, the value 0_D is used to configure SPI burst transactions with an undefined number of 32-bit data words which can be used for transactions with more than 127 data words. However, it must be ensured that the maximum number of 32-bit data words for a FIFO read is limited to 4096, so as not to exceed the maximum address range of the FIFO (0x4000 – 0x7FFF).

Afterwards, there is a two-byte wide “ADDRESS” field to define the targeted destination address. If the host controller is only capable of transferring three- or four-byte wide addresses, there is an option to extend the following wait time by several SPI clock (CLK) cycles. In such cases, the lower one or two address bytes which are sent by the host are ignored by the sensor.

Based on the type of operation, either read or write, the device must switch the direction of its used pins which requires some cycle wait time.

Once the configurable wait time is over, the last phase, called as the data phase starts. In this phase, the read or write “DATA” is transferred via one, two, or four pins, based on the selected operation mode.

Only during the data phase, if the transaction type is read and the broadcast mode is disabled, the output drivers of the respective I/O pins are enabled.

By default, the device uses MSB first, however, this can be re-configured to LSB first for all modes.

If required, the SPI transaction can be paused at any time of the protocol by stopping the SPI clock (CLK). This is possible as the device is only sensitive to SPI clock edges.

The device offers three different modes of operation:

- Standard/single SPI (DIO0 + DIO1)
- Dual SPI (DIO0 + DIO1)
- Quad SPI (DIO0 + DIO1 + DIO2 + DIO3)

The standard read mode can be used to read any 32-bit register or memory data from the device, as shown in Figure 9.

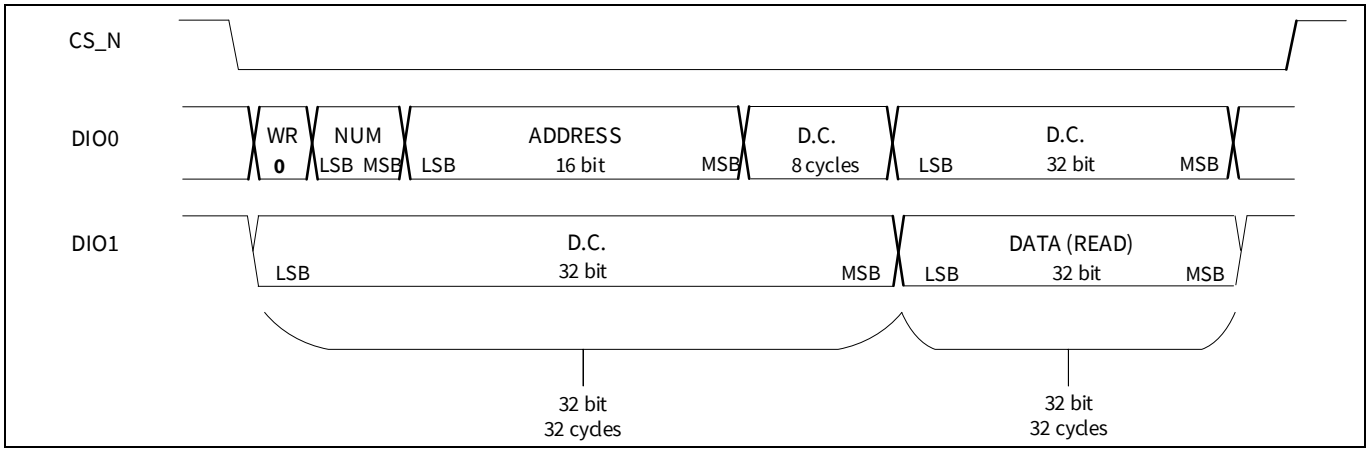


Figure 9 Standard SPI read transaction

The standard write mode can be used to write any 32-bit data to a register or a memory of the device, as shown in Figure 10.

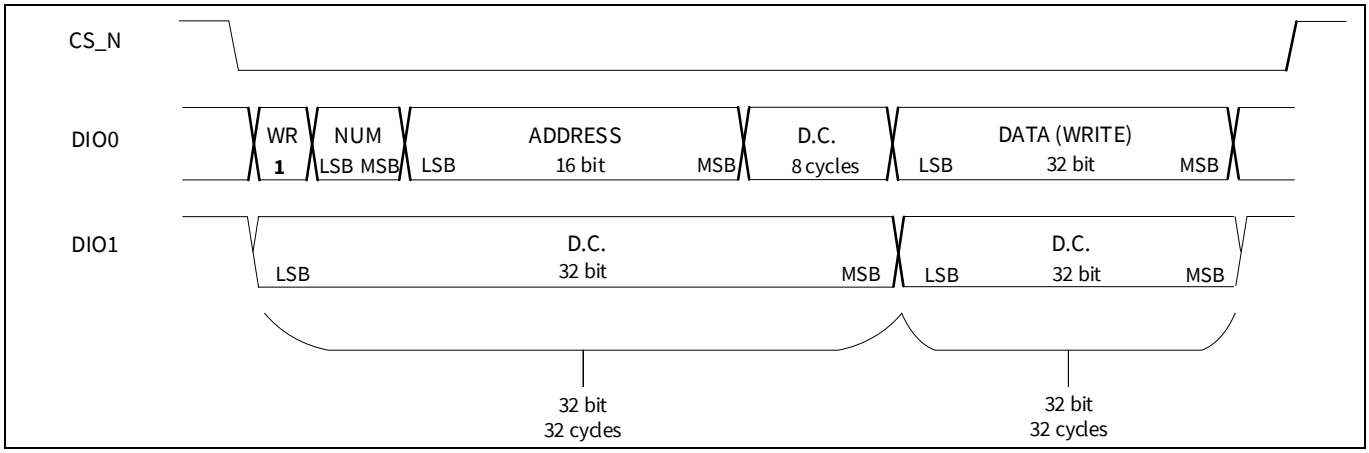


Figure 10 Standard SPI write transaction

The dual SPI mode can be used to perform a read or write access to any register or memory of the device, as shown in Figure 11.

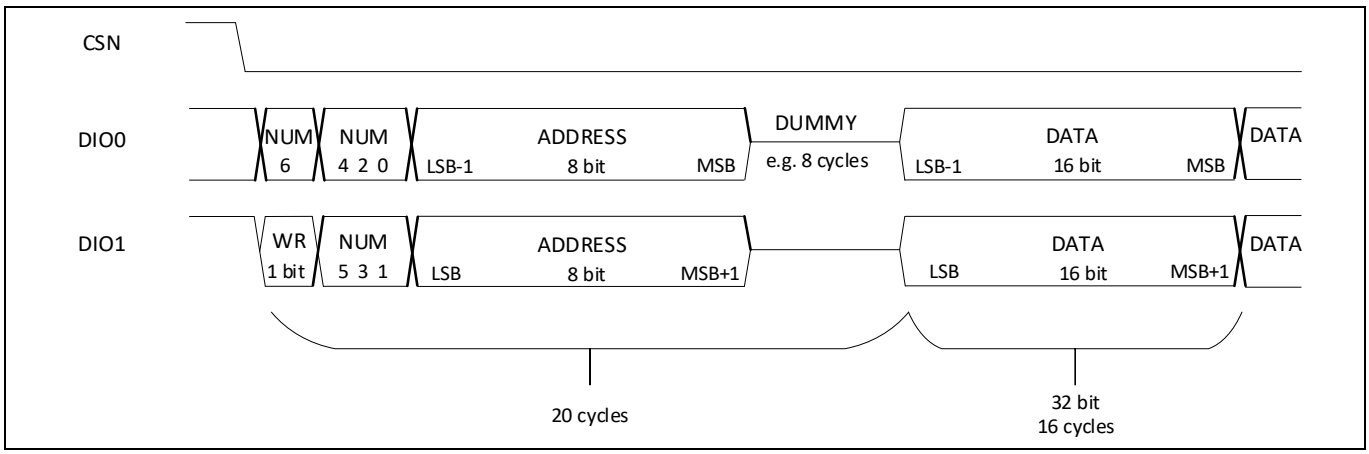


Figure 11 Dual SPI transaction

The Quad SPI mode can be used to perform a read or write access to any register or memory of the device, as shown in Figure 12.

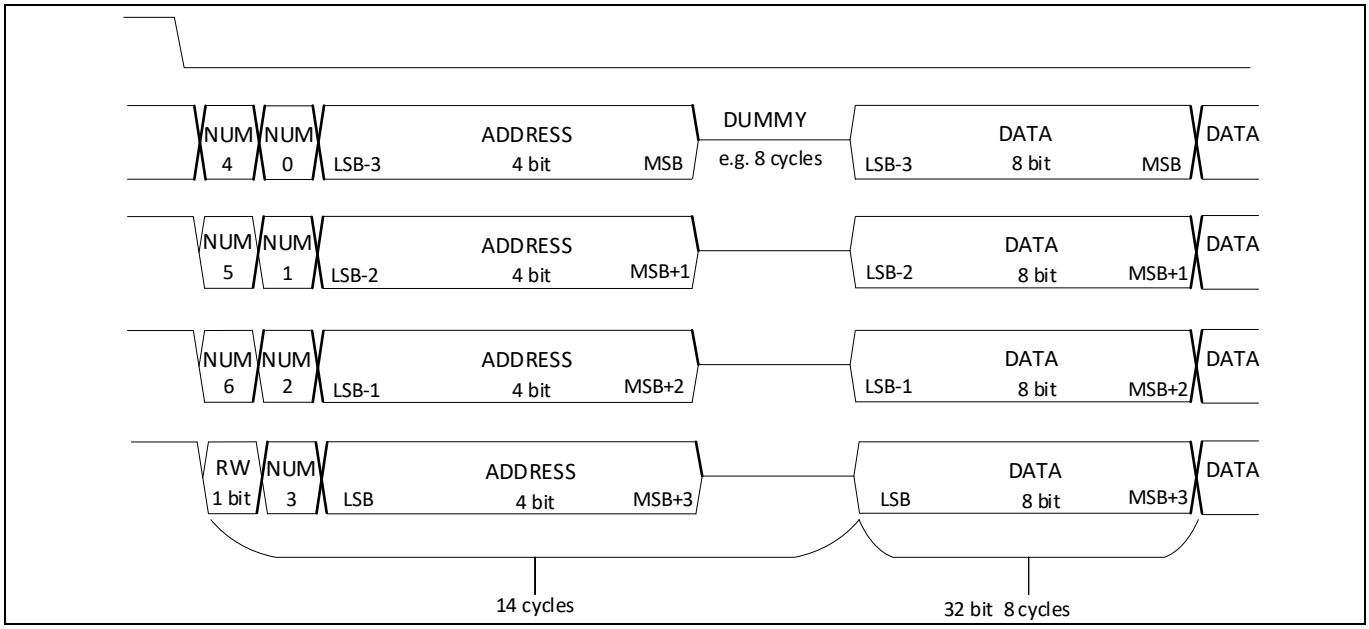


Figure 12 Quad SPI transaction

3.2.5 SPI error handling

In case the SPI protocol is violated or a SPI access to a register in the system clock is performed while the system clock is unavailable, an error bit is set. There are three types of errors:

- SPI clock number error
- SPI transaction (word) number error
- SPI no system clock error

3.2.5.1 SPI clock number error

According to the different protocols for standard SPI, dual SPI, or Quad SPI, the right number of SPI clock cycles are required while the chip select pin (CS_N) is low.

The correct number of SPI clock cycles also depends on the programmable wait time (between address and data phase) and the used SPI mode (standard, dual, and Quad).

If there are too many or too few SPI clock cycles within a SPI transaction (CS_N is low), the `CLK_STAT.SPI_CLK_NUM_ERR` bit is set. An example is shown in Figure 13.

It can be cleared by writing `1B` to the `CLK_STAT.CLR` bitfield.

Note: An SPI burst-read transaction to FIFO with an SPI clock number error could lead to a loss of data as the current data is already read from the FIFO. An SPI null command with no SPI_CLK rising edges is not considered an error condition.

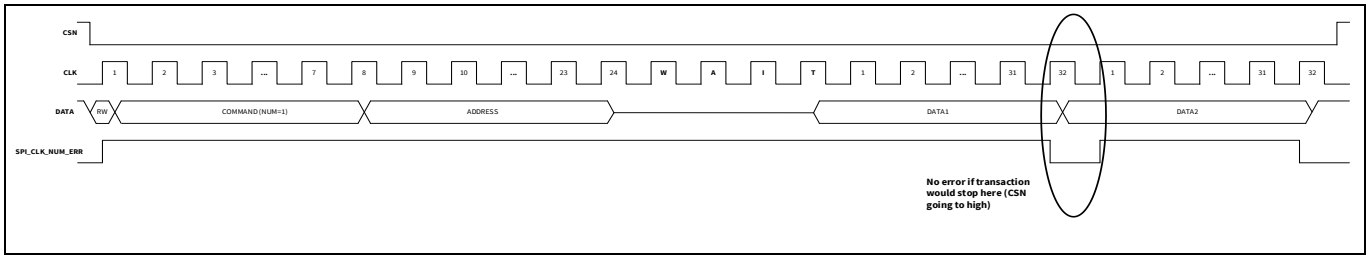


Figure 13 SPI clock number error

3.2.5.2 SPI transaction (word) number error

The first byte of the SPI protocol is defined as command phase. The NUM field reflects the number of 32-bit data words within this SPI transaction (CS_N is low).

If the SPI transaction ends before the expected number of 32-bit words is transferred or if there are too many data words within the transaction, the `CLK_STAT.SPI_TRANS_ERR` is set. An example is shown in Figure 14.

For NUM = 0 (undefined length bursts) the `SPI_TRANS_ERR` is set if the first 32-bit data word is not fully transferred, as the bare minimum for a SPI transaction is one 32-bit data word.

It is also possible that this error is set together with the `SPI_CLK_NUM_ERR`.

It can be cleared by writing `1B` to the `CLK_STAT.CLR` bitfield.

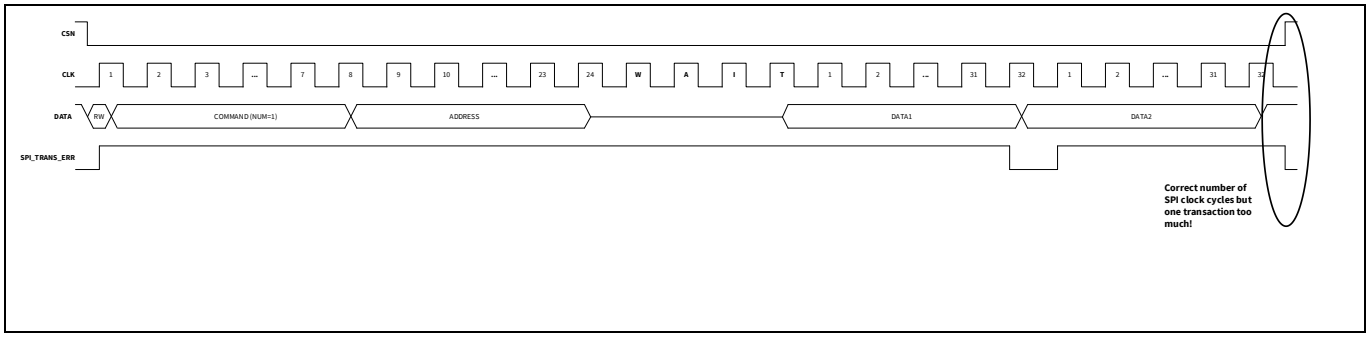


Figure 14 SPI transaction number error

3.2.5.3 SPI no system clock error

Chip architecture is defined in a way to support the lowest power consumption. If required, it is possible to disable the on-chip crystal oscillator (XOSC) manually or dynamically via `CLK_WUC.PDRQ`.

Registers in the address space up to 0x00FF can be accessed at any time and only require SPI_CLK. Whereas, registers in the address space above 0x00FF are in the system clock domain (SYS_CLK) and require an active XOSC.

If the host accesses one of these registers located in the SYS_CLK clock domain, while the system clock is not available, the SPI transaction is corrupted, and the data cannot be properly written or read which is indicated by the bit `SPI_STAT.SPI_NO_SYSCLK_ERR`.

An error is triggered as soon as the target register address is transferred (followed by a wait) and the system clock is not up and running at that time. The error can be cleared by writing `1B` to `SPI_STAT.CLR` bitfield.

In such cases, the SPI transaction must be repeated while the system clock is available.

3.3 Digital input output pins

The device comes with several digital input output (DIO) pins to reset the device, to communicate with the device via SPI, and to interact with an external host, such as trigger and interrupt.

To reduce the system cost, the device already comes with implemented level shifters in the digital I/O pads. Using the power pad VDDIO, you can apply either 1.2 V or 1.8 V based on the interface voltage on the host side.

3.3.1 Configuration

The following configuration options are available for the I/O's:

- **Enable:** Besides DIO0 and DIO1 which are required for standard SPI communication, all other DIO pins are optional and must be enabled via `IO_DIO*.EN` if required
- **I/O operation voltage:** Based on the voltage applied to the VDDIO pin, the pad driver must be configured via `IO_CTRL.IO_LVL` according to either 1.2 V or 1.8 V
- **Pull-up resistor:** There is an option to enable a weak pull-up resistor via `IO_*.PUEN`
- **Pull-down resistor:** There is an option to enable a weak pull-down resistor via `IO_*.PDEN`
- **Output driver strength:** The output is based on a push-pull configuration. Depending on the external load, the application may require selecting a different output strength via `IO_DIO*.STREN` (0_B: weak, 1_B: strong)
- **Pulse length:** In case the interrupt is configured to work in pulse mode, but the pulse duration is too short it can be extended with the settings of `IO_DIO*.PULSE_LEN` (see Section "Pulse extension")
- **Function mapping:** Due to limited numbers of input output pins and a wide range of functions, the DIO pins are shared between different functions and need to be mapped to a specific function via `IO_DIO*.MAP` based on the application requirements

Note: All outputs are in push-pull mode.

3.3.2 Mapping

To have the flexibility to use multiple functions on the same digital IO pin, the mapping of the digital I/O (DIO) to a specific function is configurable. Each DIO can select one of the following possible functions (`IO_DIO#.MAP`):

- Four internal trigger lines to trigger one of the pre-defined sequences or to start a general timer (TRIG#)
- Two additional I/Os for Quad SPI purpose. If Quad SPI `SPI_CTRL.MODE=2D (QUAD)` is selected, the device routes DIO2 and DIO3 internally to SPI to ensure proper functionality
- Two interrupt router output pins (IRQ#)
- Four digital multiplexer output pins (DMUX#)
- Two constant logic levels with the voltage of VDDIO

DIO function mapping is shown in Table 3 and Figure 15.

Table 3 DIO function mapping

MAP	DIO2	DIO3	DIO4	DIO5
0_D	TRIG0	TRIG0	TRIG0	TRIG0
1_D	TRIG1	TRIG1	TRIG1	TRIG1
2_D	QSPI2	TRIG2	TRIG2	TRIG2
3_D	TRIG3	QSPI3	TRIG3	TRIG3
4_D	IRQ0	IRQ0	IRQ0	IRQ0
5_D	IRQ1	IRQ1	IRQ1	IRQ1
6_D	DMUX0	DMUX0	DMUX0	DMUX0
7_D	DMUX1	DMUX1	DMUX1	DMUX1
8_D	DMUX2	DMUX2	DMUX2	DMUX2
9_D	DMUX3	DMUX3	DMUX3	DMUX3
10_D to 13_D	RSVD	RSVD	RSVD	RSVD
14_D	LOW	LOW	LOW	LOW
15_D	HIGH	HIGH	HIGH	HIGH

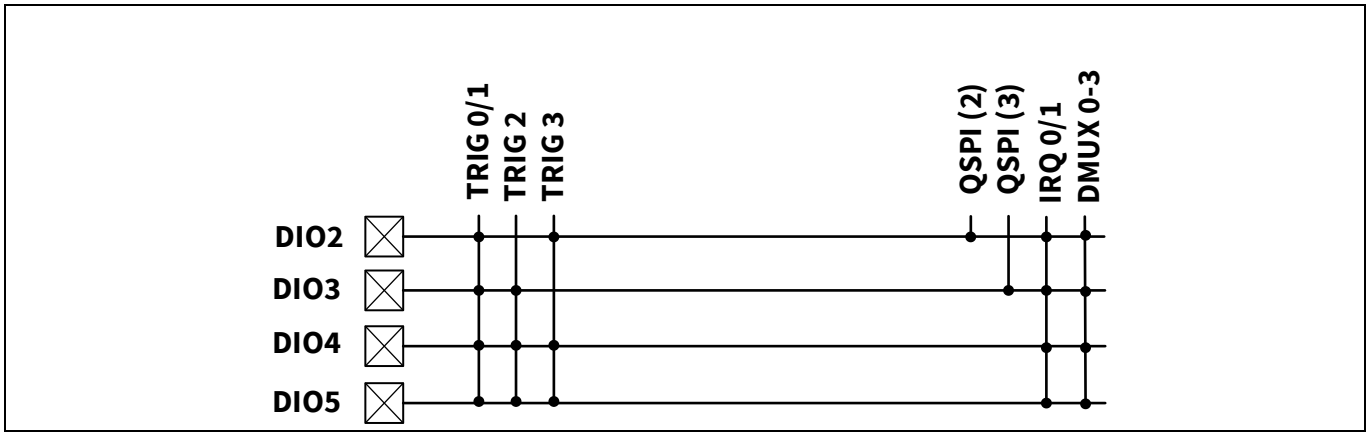


Figure 15 DIO function mapping

Several internal signals can be routed via the DMUX to a digital I/O pin to be observed outside of the sensor. For that the DMUX selection can be configured via `IO_CTRL.DMUX#_SEL`, accordingly.

DMUX# mapping list is provided in Table 4 and Table 5.

Table 4 DMUX mapping of DMUX0 and DMUX1

Selection	Name	Description
0_D	PLL_RAMP_START	PLL: ramp start
1_D	PLL_RAMP_PROGRESS	PLL: ramp progress
2_D	PLL_ACTIVE	PLL: active
4_D	PLL_RESET	PLL: reset
5_D to 6_D	RSVD	Reserved
7_D	SYS_CLK	FD: frequency doubler clock out
8_D	ADC_RESET	ADC: reset
9_D	ADC_EN	ADC: module enable
10_D to 12_D	RSVD	Reserved
13_D	ADC_START	ADC: start trigger
14_D	ADC_EOC	ADC: end of conversion
15_D	CHIRP_ACTIVE	SEQ/FSM: chirp active
16_D	ABB_RESET	ABB: reset
17_D	PA_ACTIVE	RF: PA active
18_D to 19_D	RSVD	Reserved
20_D	PDC_UPDN	PDC: domain powered up/down (status)
20_D to 23_D	RSVD	Reserved
24_D	RCOSC	RCOSC: clock out
25_D	XOSC_READY	XOSC: module ready
26_D	MD_DONE	MD: algorithm done (set after each post-processing state and cleared when next post-processing state is started)
27_D	MD_EXEC	MD: algorithm execution starts (start trigger of each post-processing step)
28_D	SEQ0_ACTIVE	SEQ: sequence0 active
29_D	SEQ1_ACTIVE	SEQ: sequence1 active
30_D	SEQ2_ACTIVE	SEQ: sequence2 active
31_D	SEQ3_ACTIVE	SEQ: sequence3 active

Table 5 DMUX mapping of DMUX2 and DMUX3

Selection	Name	Description
0_D	SADC_EN	SADC: module enable
1_D	SADC_START	SADC: start trigger
2_D	SADC_EOC	SADC: end of conversion
3_D	RSVD	Reserved
4_D	PLL_LOCK	PLL: lock
5_D to 8_D	RSVD	Reserved
9_D	FIFO_TH	FIFO: threshold reached
10_D	FIFO_PWR	FIFO: SRAM powered up
11_D	RX_BUF_PWR	RX_BUF: SRAM powered up
12_D to 13_D	RSVD	Reserved
14_D	LFSR1_OUT	LFSR: LFSR1 bit 0 output (BPSK)
15_D	LFSR2_OUT	LFSR: LFSR2 bit 0 output (BPSK)
16_D	GTMR0	GTMR: Timer0 output trigger
17_D	GTMR1	GTMR: Timer1 output trigger
18_D	GTMR2	GTMR: Timer2 output trigger
19_D	GTMR3	GTMR: Timer3 output trigger
20_D	MD_ULP_EVENT	MD: Ultra-low power-based motion event
21_D	MD_RD_EVENT	MD: Range Doppler-based motion event
22_D to 27_D	RSVD	Reserved
28_D	SEQ0_DONE	SEQ: sequence0 done/over pulse
29_D	SEQ1_DONE	SEQ: sequence1 done/over pulse
30_D	SEQ2_DONE	SEQ: sequence2 done/over pulse
31_D	SEQ3_DONE	SEQ: sequence3 done/over pulse

3.3.3 Pulse extension

Some signals may only be active for one single system clock cycle which is 12.5 ns at 80 MHz.

To relax the timing requirements for such fast signals at the external hardware (host side), there is an option to prolong these signals by several clock cycles (`IO_DIO#.PULSE_LEN`) on the sensor side, as shown in Table 6 and Figure 16. If a new pulse is generated while the previous one is still extended, the pulse stretching counter is re-started with the last generated pulse.

Note: This functionality is only available for DMUX and IRQ in EDGE mode. For all other cases (DIO# mapped to QSPI, TRIG, IRQ (LEVEL)), the PULSE_LEN must be configured to 0.

Table 6 DIO pulse extension

PULSE_LEN	Name	Description
0_D	None	Bypass
1_D	LEN2	2 clock cycles
2_D	LEN4	4 clock cycles
3_D	LEN8	8 clock cycles
4_D	LEN16	16 clock cycles
5_D	LEN32	32 clock cycles
6_D	LEN64	64 clock cycles
7_D	LEN128	128 clock cycles

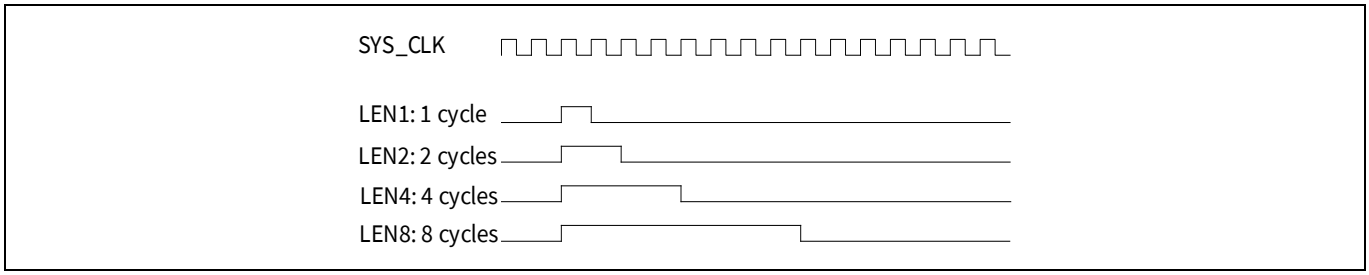


Figure 16 DIO pulse extension

3.3.4 Trigger

The "Mapping" Section shows that the different trigger lines (TRIG0-3) can be used by different modules like the general timer (GTMR0-3) or the sequencer (SEQ) to trigger pre-defined sequences (SEQ0-3). However, to detect the incoming triggers properly, the length of a pulse must be at least 25 µs (two times T_{RCOSC}), as shown in Figure 17.

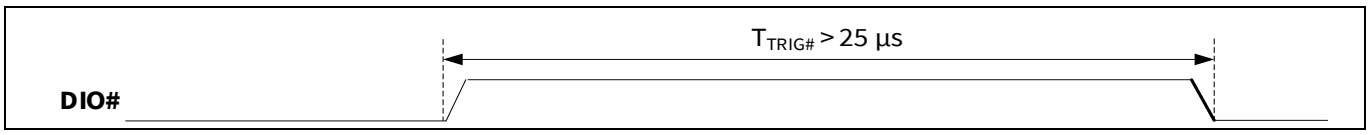


Figure 17 External trigger pulse length

Depending on which internal oscillator is active (XOSC or RCOSC) when the TRIG pulse is received, a different latency can be observed. If the device is in one of the sleep modes (SLEEP, DEEPSLEEP), the 80 MHz system clock (SYS_CLK) is off. If only the 80 kHz RCOSC is running, there will be a delay of at least one RCOSC clock cycle before an internal trigger pulse is generated to start the wakeup sequence. In contrast, when SYS_CLK is active, only a short latency (at least one SYS_CLK cycle) is observed until the sequence starts. This is shown in Figure 18.

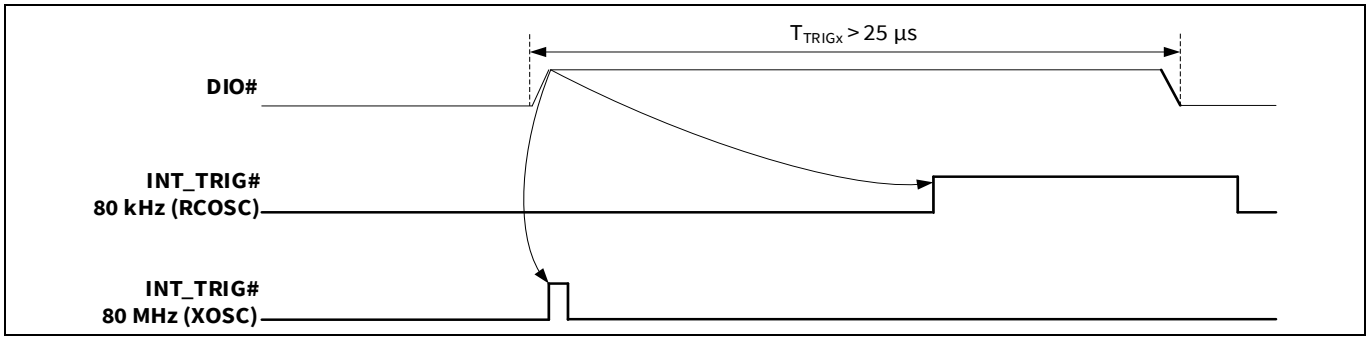


Figure 18 Trigger latency of RCOSC and XOSC

Note: Whenever an external trigger is used, a certain order must be ensured. Enable the DIO pad first, map it to the related trigger, and finally enable the destination (sequence or GTMR#).

4 Device initialization

To start the device properly, several steps must be performed (see Figure 19). Some steps need to be executed sequentially, whereas others can run in parallel, as they are performed by the hardware and do not require any interaction with the host (software).

Depending on the application requirements, some steps may be optional. In such cases, these steps can be skipped and do not need to be executed.

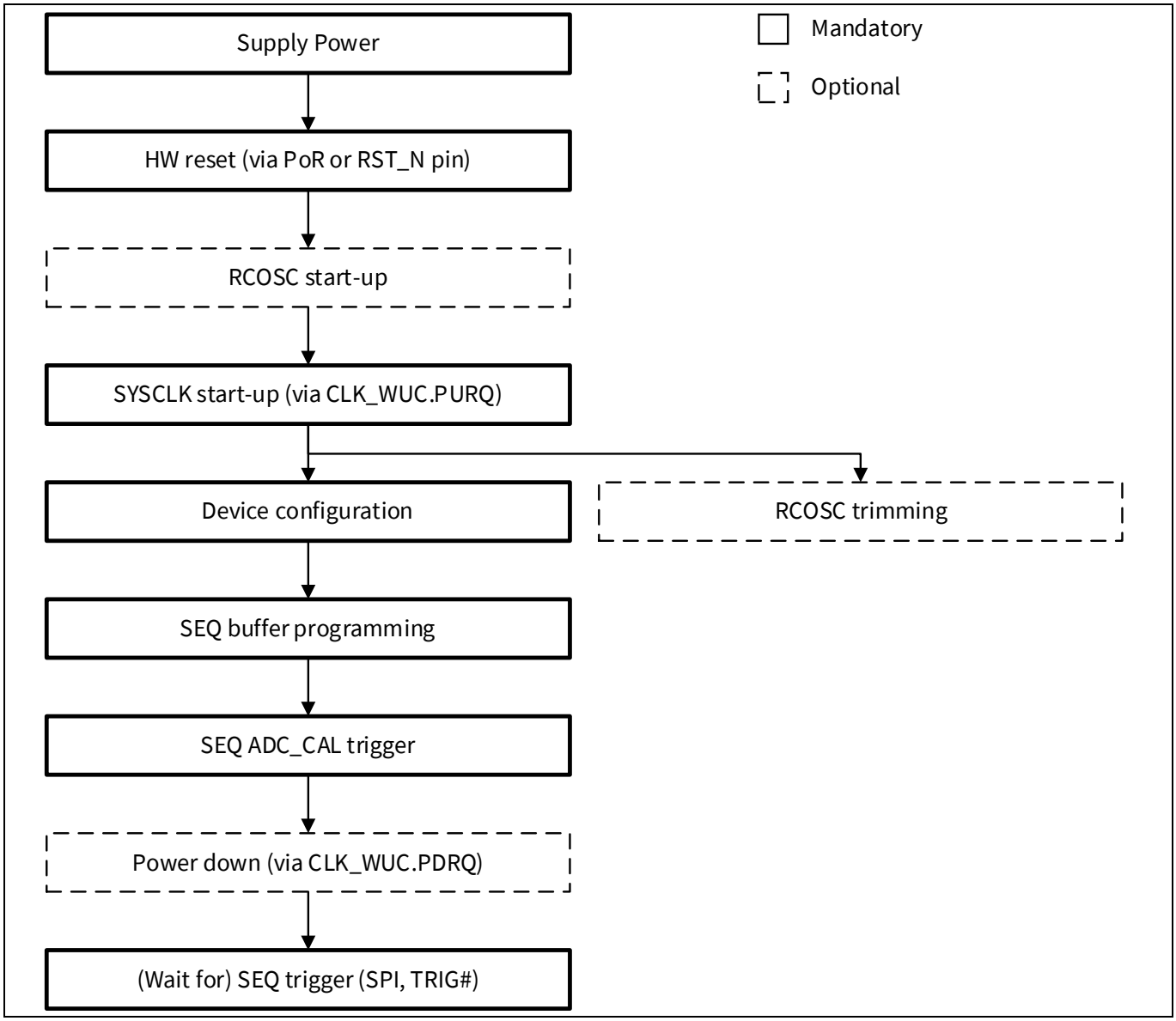


Figure 19 Device initialization

Note: Any interaction with the device, like reading and writing, requires a running system clock provided via XOSC module. The only exceptions are interactions with clock (CLK_*) and SPI (SPI_*) registers to enable communication with the device and startup the system clock properly.

4.1 Power supply startup

To operate the XENSIV™ BGT60CUTR13AIP device in a fully functional mode, all power supplies (VDDx) must be applied.

To avoid potential high current for a few microseconds (µs), the digital supply VDDD should be powered up first, as shown in Figure 20. Subsequently, all other domains (including VDDIO) can be powered up.

There are no functional and reliability issues if all power domains are powered up simultaneously. However, there could be a potential increase in current consumption for a few microseconds (µs) when VDDIO is applied before VDDD is fully available. Since the ENPS (enable power supply) signal of the digital pads is controlled by the digital power domain VDDD, it is crucial that VDDD is available before the other domains to prevent temporary increase in current consumption.

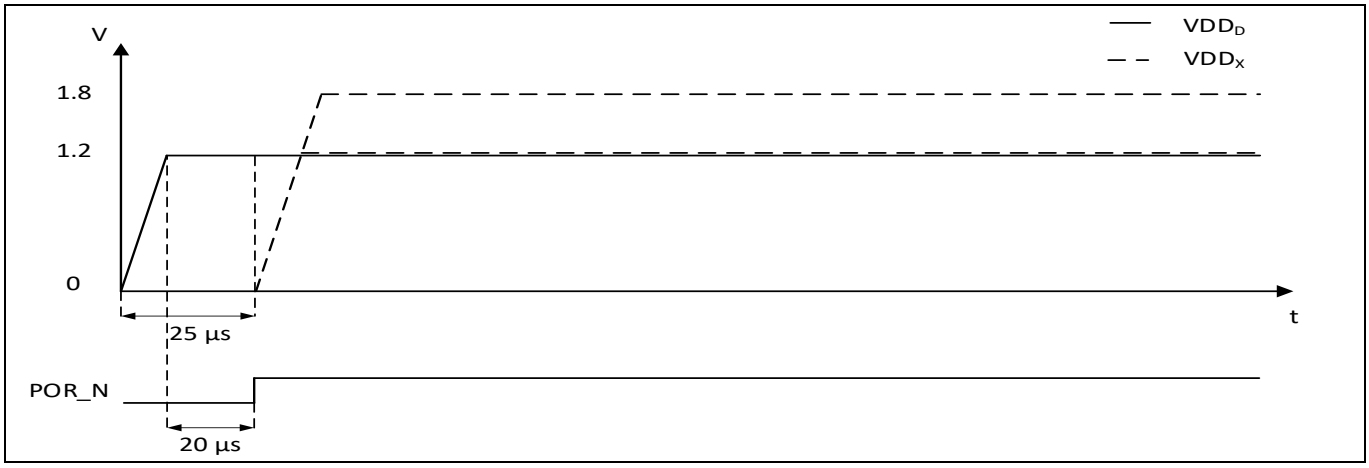


Figure 20 Power supply startup

4.2 Reset and clock startup

After each power supply is properly powered up and available, the device must be reset to enter a defined state, and the system clock must be enabled.

4.2.1 Hardware and power-on reset (POR)

Once the VDDD power supply is settled, a power-on-reset (POR) is released. Alternatively, an external reset (RST_N) can be performed to set the device to its default state. For the RST_N, a high → low → high sequence must be applied to the RST_N pin for the defined reset duration, as shown in Figure 21. Both POR and RST_N are low-active and are internally combined using a logical AND operation to generate the hardware reset (HW_RST). Consequently, the reset release time is determined by the last reset signal to be released.

Note: Without HW_RST reset, the device is in an undefined state, and proper SPI communication cannot be guaranteed including a software reset (SW_RST) request.

- Time reset T_{RES}: 500 ns
- Time chip select (CS) after reset (T_{CS_ARES}): 5 µs (minimum)

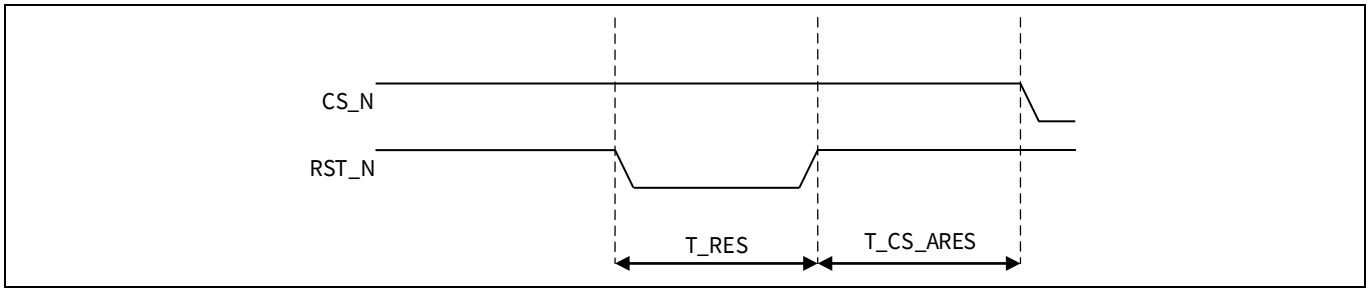


Figure 21 Hardware reset sequence

4.2.2 Bootloader (BL)

4.2.2.1 Functionality

The device can be used together with a host controller, which communicates via the Serial Peripheral Interface (SPI) and the digital input/output pins (DIO). In such a setup, the configuration of the chip can be done via the SPI.

Alternatively, the device can be operated in a standalone mode. In such a setup, the reset values of the configuration registers can be overwritten by automatically loading one of up to three predefined configurations from the embedded read-only memory (ROM).

The required configuration from Table 7 can be selected by setting the pins DIO2 and DIO3 to a logical low or logical high voltage level. If the loading process should be skipped, a high level must be applied to DIO2 and DIO3 pins. If the device is controlled by an external host via Single-SPI, the DIO2 and DIO3 pins can be left floating. These pins have “weak” internal pull-ups which disable the bootloader functionality. In case the use case requires Quad SPI, these pins need to be connected to the external host.

After the hardware reset is released, the RCOSC is started, and the bootloader executes its startup procedure.

The pads DIO2 and DIO3 are enabled along with their pull-ups, and after a hard-coded number of RCOSC clock cycles (`BOOT_STAT.SENSE_DEL`), their logical values are sensed. The sequence is as follows:

1. Release HW_RST (power-on-reset (POR) or RST_N pin)
2. Enable DIO2 and DIO3 pads together with their pull-ups
3. Wait for BOOT_STAT.SENSE_DEL
4. Sense DIO2 and DIO3 inputs
5. Disable DIO2 and DIO3 pins and their pull-ups

Table 7 DIO bootloader configurations

DIO3	DIO2	Comment
0	0	Bootloader enabled (CFG0)
0	1	Bootloader enabled (CFG1)
1	0	Bootloader enabled (CFG2)
1	1	Bootloader disabled

As a part of the BL sequence, the RC oscillator (RCOSC) trimming is performed.

The status for the BL can be read from the ROM status register:

`BOOT_STAT.CFG`

- 0: CFG0 Bootloader enabled and configuration 0 selected
- 1: CFG1 Bootloader enabled and configuration 1 selected
- 2: CFG2 Bootloader enabled and configuration 2 selected
- 3: DISABLED Bootloader disabled

`BOOT_STAT.STATE`

- 0: INACTIVE
- 1: RSVD
- 2: DONE
- 3: ERROR

After 5 ms following the reset release, the bootloader execution is complete, and the host can start to trigger a sequence or to configure the device. If the bootloader is disabled, only 1 ms wait time is required.

4.2.3 RC oscillator (RCOSC)

To trigger a sequence periodically or by an external trigger (TRIG0-3) while the chip is in Sleep or DeepSleep mode (XOSC is off and SYS_CLK is not available), the embedded low-power RC oscillator must be enabled. The 80 kHz on-chip RCOSC is used to clock the general timer (GTMR) module, the trigger (TRIG) receiving module, and the XOSC wakeup controller (WUC).

The RCOSC startup step is optional and is only required in the following cases:

- The general timer module (GTMR) is used to trigger (start) sequences by the device itself without any interaction with the external host microcontroller
- Any sequence can be started via an external trigger while the chip is in Sleep or DeepSleep mode (XOSC is off and SYS_CLK is not available)
- Powering up of the embedded XOSC shall be done by the device itself via `CLK_WUC.PURQ`

If at least one of the above scenarios is required, the low-power RC oscillator must be enabled. The RC oscillator can be configured via `CLK_RCOSC_CTRL.MODE` to operate in one of the following modes:

- 1_D (ON): the RCOSC is always running
- 2_D (DYN/AUTO): the RCOSC is automatically enabled and disabled by the device itself. During an active sequence, the RCOSC is disabled by the device to avoid any spurs in the intermediate frequency (IF), and enabled once the sequence is over

Independent of the mode used, the RCOSC must be trimmed to match the targeted 80 kHz frequency. This trimming is required after RCOSC startup or if the voltage or temperature is changing over time. See Section “RC oscillator trimming”.

The status of the RCOSC and the RCOSC_CLK can be checked by reading the bitfield `CLK_STAT.RCOSC_STATE` (0_B: INACTIVE, 1_B: ACTIVE).

4.2.4 XTAL oscillator (XOSC)

To operate the device, a system clock of 76.8 MHz or 80 MHz is required. This clock can either be generated by the embedded XTAL oscillator or via an external clock. For both cases, the XOSC module must be enabled and the required mode to be set via `CLK_XOSC_CTLR0.MODE`. In case the system clock is generated with the embedded XTAL oscillator, an external XTAL and two external capacitors must be connected to the XI and XO pin.

4.2.4.1 Manual software startup

The device offers the option to manually control the SYS_CLK by powering up and down the XOSC module by software via SPI transactions. For that, the XOSC mode must be set to `MODE = 1`$_D$ (MAN/SW). With this setting, the system clock is completely under the control of the host controller which must ensure that the XOSC module is up and running when a sequence must be executed or data needs to be analyzed by the embedded digital signal processor (DSP).

The status of the XOSC and the SYS_CLK can be checked by reading the bitfield `CLK_STAT.XOSC_STATE` (0_B: INACTIVE, 1_B: ACTIVE).

The correct XOSC power-up timing, shown in Figure 22, must be ensured by the host and should be controlled via the register `CLK_XOSC_CTRL0`. The power-down does not need any special sequential order and can be performed by setting all XOSC control bits to zero.

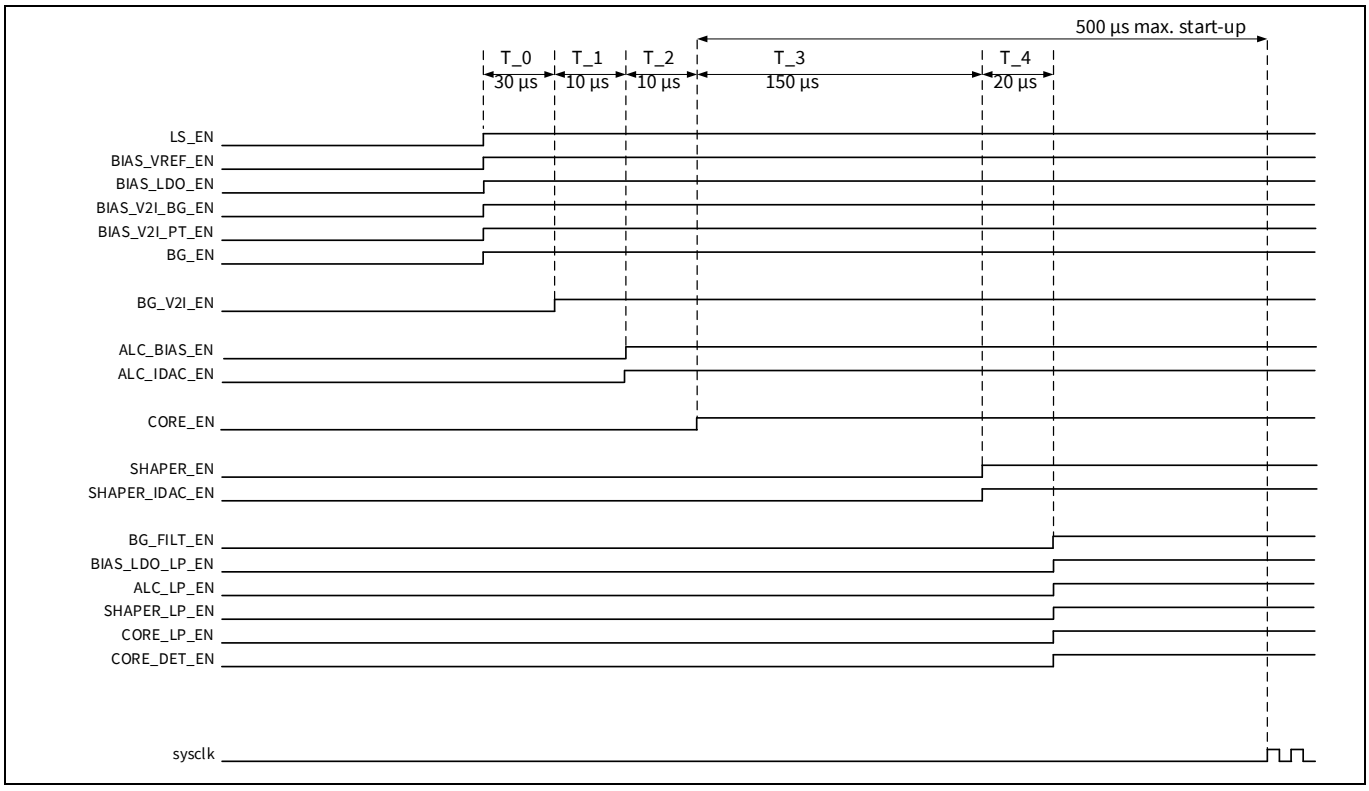


Figure 22 Manual software XOSC startup timing (internal)

4.2.4.2 Automatic hardware startup

To handle the startup and power down of the embedded crystal oscillator by the device itself, the XOSC operation mode must be set to `MODE = 2D` (AUTO/HW). With this setting, the system clock (SYS_CLK) is dynamically enabled and disabled by the hardware. In cases where sequences (frames) must be executed or acquired data must be processed using the embedded digital signal processor (DSP), the XOSC is powered up via the wakeup controller (WUC). Furthermore, there is also an option to request power-up with the WUC module via software (`CLK_WUC.PURQ`). To reduce the overall power consumption, the XOSC module is powered down if the sequence and the post-processing of the data is over.

Note: Ensure that the RCOSC is enabled to clock the WUC module.

The following timer settings must be set for an automatic XOSC startup in the WUC:

Code listing 1 Automatic hardware XOSC startup timing

```
CLK_XOSC_CTRL1.TR_0 = 3D
CLK_XOSC_CTRL2.TR_1 = 1D
CLK_XOSC_CTRL2.TR_2 = 1D
CLK_XOSC_CTRL2.TR_3 = 12D
CLK_XOSC_CTRL2.TR_4 = 2D
```

The overall startup time requires maximum 550 µs. The status of the XOSC and the SYS_CLK can be checked by reading the bitfield `CLK_STAT.XOSC_STATE` (0_B: INACTIVE, 1_B: ACTIVE).

4.2.5 External oscillator (EXT_OSC)

In addition to using the embedded crystal oscillator to generate a clock, there is also an option to use an already existing clock that is available in the system. In that case, the XOSC module is also re-used. However, the core and driver components responsible for driving the XTAL remain disabled.

4.2.5.1 Manual software startup (external clock)

The manual startup of the system clock can be done in a similar way as already described in Section "Manual software startup". For this use case, the XOSC mode must be set to `MODE = 4D` (MAN_EXT / SW_EXT). With this setting, the system clock is completely under the control of the host controller which must ensure that the XOSC module is up and running when a sequence needs to be executed or data needs to be analyzed by the embedded digital signal processor (DSP).

The status of the XOSC and the SYS_CLK can be checked by reading the bitfield `CLK_STAT.XOSC_STATE` (0_B: INACTIVE, 1_B: ACTIVE).

The correct XOSC power-up timing, shown in Figure 23, must be ensured by the host and should be controlled via the register `CLK_XOSC_CTRL0`. The power-down does not need any special sequential order and can be performed by setting all XOSC control bits to zero.

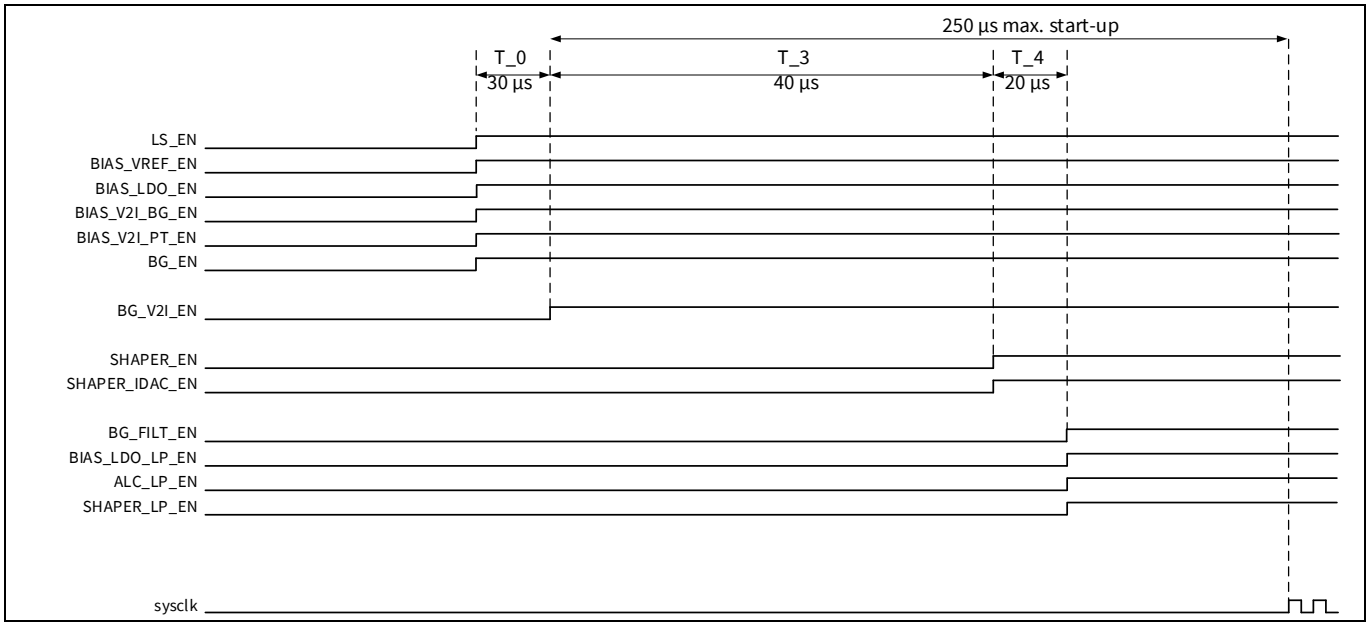


Figure 23 Manual software XOSC startup timing (external clock)

4.2.5.2 Automatic hardware startup (external clock)

To handle the startup and power down of the embedded crystal oscillator by the device itself, the XOSC operation mode must be set to `MODE = 3D` (AUTO_EXT / HW_EXT). With this setting, the system clock (SYS_CLK) is dynamically enabled and disabled by the hardware. If sequences (frames) need to be executed or acquired data needs to be processed with the embedded digital signal processor (DSP), the XOSC is powered up via the wakeup controller (WUC). Furthermore, there is also an option to request power-up with the WUC module via software (`CLK_WUC.PURQ`). To reduce the overall power consumption, the XOSC module is powered down once the sequence execution and data post-processing are complete.

Note: Ensure that the RCOSC is enabled to clock the WUC module.

The following timer settings must be set for an automatic XOSC startup in the WUC:

Code listing 2 Automatic hardware XOSC startup timing (external clock)

```
CLK_XOSC_CTRL1.TR_0 = 3D
CLK_XOSC_CTRL2.TR_1 = 0D
CLK_XOSC_CTRL2.TR_2 = 0D
CLK_XOSC_CTRL2.TR_3 = 3D
CLK_XOSC_CTRL2.TR_4 = 2D
```

4.2.6 Frequency doubler

The device includes an integrated frequency doubler (FD) for both the system clock (SYS_CLK) and the PLL reference clock (PLL_REF_CLK). This enables the use of a cost-effective 38.4 MHz or 40 MHz crystal for the XOSC or enables the PLL to operate in a high-performance mode at 153.6 MHz or 160 MHz. The frequency doubler module receives its input clock from the XOSC module and must be enabled via `CLK_FD.EN` to provide the system clock (SYS_CLK) to the device.

Note: The FD module must be enabled to provide the SYS_CLK to the device as it is used for global chip clock gating.

There are four operating modes available to cover different scenarios:

- MODE0: there is no frequency doubling and no duty cycle correction applied
- MODE1: frequency doubling is enabled; however, no duty cycle correction applied
- MODE2: frequency doubling is active and input/output duty cycle correction is applied
- MODE3: frequency doubling is active and input/output duty cycle correction is applied only to SYS_CLK but not to PLL_REF_CLK

The digital logic of this device is synthesized for a maximum operating frequency of 80 MHz. Therefore, when using a 76.8 MHz or 80 MHz input clock, it must be ensured that only the PLL reference clock (PLL_REF_CLK) is doubled, while the system clock (SYS_CLK) is bypassed. This can be done by selecting the right frequency via `CLK_FD.SRC_FREQ_SEL`.

Note: The frequency doubler must be configured prior to the XOSC module startup to prevent clock glitches.

If no clock frequency doubling is required, MODE0 can be selected. MODE2 and MODE3 offer the feature to correct the duty cycle, as the frequency doubler is based on simple XOR functionality. This is necessary as the internal system clock has a duty cycle requirement of 50% ±10 %.

For these two modes, `CLK_FD.DCC_IN_SEL` and `CLK_FD.DCC_OUT_SEL` are available to modify the duty cycle. If the input duty cycle is already too high and needs to be lowered, a clock inversion is required via the bit `CLK_FD.CLK_INV_EN`. After that, it is possible to increase the duty cycle of the clock by introducing additional delay to one XOR input.

Figure 24, Figure 25, Figure 26, and Figure 27 show how the frequency doubler duty-cycle correction is applied.

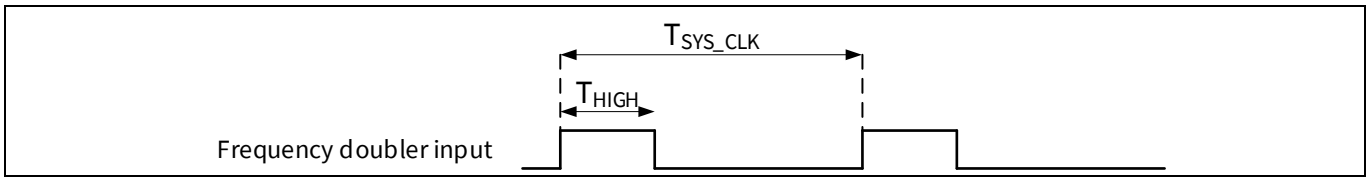


Figure 24 Frequency doubler input clock

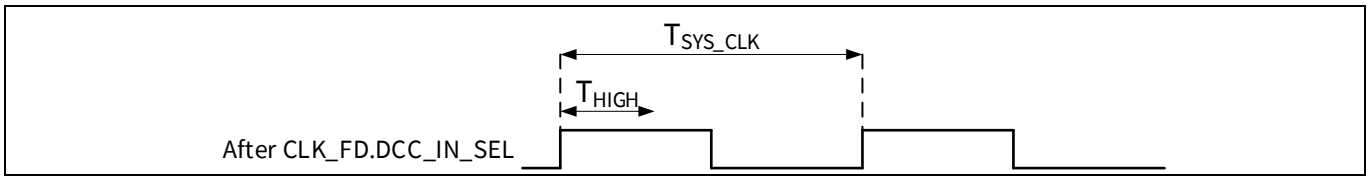


Figure 25 Frequency doubler after input clock duty-cycle correction

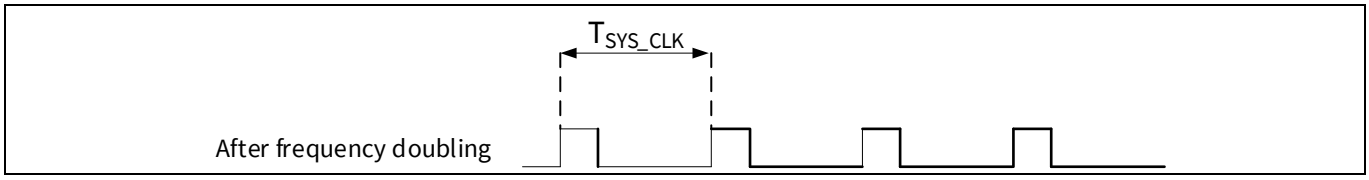


Figure 26 After frequency doubling

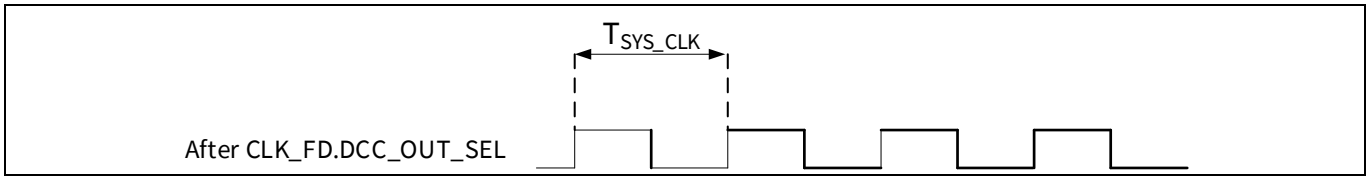


Figure 27 Frequency doubler after output duty cycle correction

4.3 Trimming and calibration

4.3.1 RC oscillator trimming

The RC oscillator trimming can be done by hardware or software and needs to be configured via `CLK_RCOSC_TRIM.TRIM_MODE`.

4.3.1.1 Automatic hardware trimming

The device has an embedded RC oscillator trimming algorithm (approximation) that can be used to perform the RCOSC trimming after reset or in response to voltage and temperature changes. For this, both the RCOSC and XOSC must be active, and the following need to be configured.

Code listing 3 Automatic RC oscillator trimming

```
CLK_RCOSC_TRIM.TRIM_MODE = 1 // automatic hardware trimming
CLK_RCOSC_TRIM.TRIM_CYCLE = 0 to 2 //RCOSC cycles +1
CLK_RCOSC_TRIM.TRIM_TRIG = 1 //start with separate SPI command
```

The hardware performs the trimming and applies the resulting value according to the RC oscillator. Additionally, you can read back the trimming value from the device 1.5 ms after trimming is initiated by reading `CLK_RCOSC_TRIM.TRIM_VALUE`.

4.3.1.2 Manual software trimming

Alternatively, there is an option to trim the RC oscillator manually via software based on its own user-specific algorithm. For this, both RCOSC and XOSC must be active and the following need to be configured.

Code listing 4 Manual RC oscillator trimming

```
CLK_RCOSC_TRIM.TRIM_MODE = 0 // manual software trimming
CLK_RCOSC_TRIM.TRIM_CYCLE = 0 to 7 // RCOSC cycles +1
CLK_RCOSC_TRIM.TRIM_VALUE = 0 to 31
CLK_RCOSC_TRIM.TRIM_TRIG = 1 //start with separate SPI command
```

After the trimming was triggered, the hardware counts the number of XOSC clock cycles based on the configured number of RCOSC cycles (`CLK_RCOSC_TRIM.TRIM_CYCLES`). If the counter reaches its maximum, the counter is stopped (no roll-over).

Now, the host must read back the counter result via `CLK_RCOSC_TRIM.TRIM_CNT` and based on its value, the host can reprogram the trimming value with a separate SPI command. This can be repeated several times until the RCOSC period matches the expected counter value. RCOSC trimming cycle is shown in Figure 28.

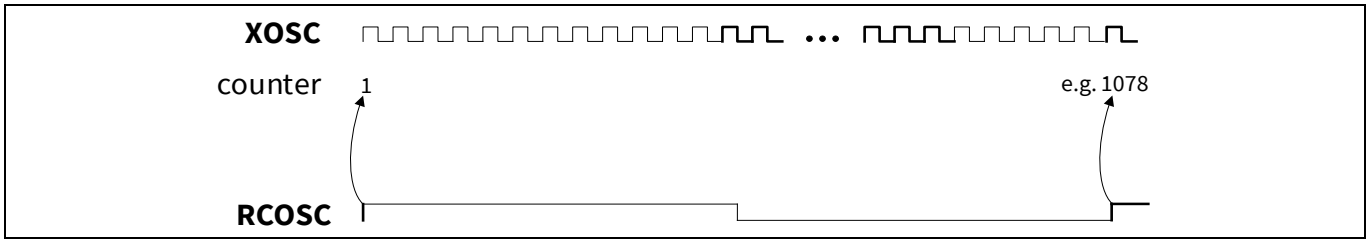


Figure 28 RCOSC trimming cycle

4.3.2 ADC calibration

There are four ADCs integrated, one for each receiver (RX) channel to convert the intermediate frequency (IF) and one dedicated sensor ADC (SADC) to convert analog voltages into digital representation. The SADC is used to measure the duty cycle of the system clock, the chip temperature, and the power levels on the transmitter (TX) and receiver (RX) channels via a peak-detector circuit.

To get full performance, the multi-ADC (MADC) module must perform an offset, linearity, and timing calibration. This is done by loading the ADC calibration sequence (ADC_CAL_SEQ) to the sequence buffer memory (SEQ_BUFFER) and executing it by triggering the associated sequence (SEQ#). There are up to four free configurable sequences (SEQ0 – SEQ3) available, each with a dedicated pointer representing the offset start address of the sequence inside the sequence SEQ_BUFFER.

In the following example, the sequence 0 (SEQ0) is used; however, any other available sequence can also be used. The sequence buffer memory is located at address `0x3000` and the offset is set to `0x0300` (`SEQ0_CTRL.START_ADDR_OFFSET = 300`$_H$). After the ADC_CAL_SEQ is loaded to the SEQ_BUFFER, the sequence can be triggered and executed with `CLK_WUC.SEQ0_TRIG = 1`$_B$ and will take ~170 µs.

Based on the application requirements, the system clock (SYS_CLK) can either remain active or be disabled, allowing the device to enter the lowest power state, DeepSleep, after the ADC calibration sequence is completed. This can be configured via `CLK_WUC.SEQ#_DONE` (0_D: DeepSleep, 2_D: ACTIVE).

Code listing 5 ADC calibration sequence (ADC_CAL_SEQ)

```
// Start-up modules (45 us) and execute calibration (125 us)
   0x3300 0x30F10026
   0x3304 0x3489008E
   0x3308 0x30F10066
   0x330C 0x018E01E6
   0x3310 0x03E63051
   0x3314 0x307D1FFE
   0x3318 0x549D1FE6

// Power-down modules (0.1 us)
   0x331C 0x03E6000A
   0x3320 0x0031018E
   0x3324 0x008E0006
   0x3328 0x000E0031
   0x332C 0x00040004
```

The resulting calibration values are stored in the always-on power domain of the device to keep them as long as the digital power domain (VDDD) is supplied and no reset is performed. These ADC calibration values are automatically applied to the ADC each time an ADC is powered up.

Note: The ADC calibration sequence must be executed at least once after a hardware (HW_RST) or software reset (SW_RST) is executed. This is also the case when the hardware reset is executed via the embedded power-on reset (PoR) when the digital supply VDDD is applied.

4.3.3 FD duty cycle trimming

To adjust the duty cycle of the system clock, the embedded sensor ADC can be used to measure the duty cycle. For this `SADC_IDX#_CTRL.SEL = 0D` (DC) must be selected and a sensing chirp must be triggered. It is also recommended to enable decimation (`SADC_IDX#_CTRL.DF = 3D`) to get a better average value. Depending on which signal is of interest, different signals at the frequency doubler can be routed to the sensor ADC input (`CLK_FD.DCMMUX_SEL`). The following can be selected:

- XTAL: input clock of XTAL
- IN: input duty cycle corrected XTAL clock
- OUT: output duty cycle corrected clock to digital (SYS_CLK)
- REF_HIGH: reference voltage high
- REF_LOW: reference voltage low
- CLK_DIV2: system clock divided by 2

A perfect 50% duty cycle would result in a voltage (REF_HIGH + REF_LOW) / 2. The duty cycle correction can be done in incremental steps starting with the lowest value of `DCC_IN_SEL` and `DCC_OUT_SEL`. It is necessary to first correct the input duty cycle (`DCC_IN_SEL`) before the output duty cycle (`DCC_OUT_SEL`) can be corrected.

If the input duty cycle is already too high (> 50%), the input clock must be inverted first, as the duty cycle can only be increased and not be decreased. This inversion can be enabled via the `CLK_INV_EN` bit. However, the system clock must be disabled to prevent glitches on the system clock tree.

Procedure:

1. Disable the clock
2. Setting `CLK_FD.CLK_INV_EN = 1B`
3. Enable the clock

4.4 Configuration

To use the device, it must be configured according to the application requirements. That is done by programming the device register according to the register descriptions. All registers can be programmed via SPI with 32-bit data and are accessible on the address map from `0x000` to `0x1FFF`.

4.4.1 Frame configuration

All frames that are defined as sequences and are executed during normal device operation need to be loaded to the sequence buffer (SEQ_BUFFER) memory. For details about possible chirp configurations and frame definitions, see Section “Radar frame definition”.

The memory has a size of 1 KibiByte (KiB) and can store up to 512 commands (each command consists of two bytes) which are interpreted by the frame sequencer. Basically, any number of sequences can be loaded to the SEQ_BUFFER memory if the overall memory size is not exceeded. However, there are only four sequence start address pointers available (SEQ0 – SEQ3). If more sequences are stored in the SEQ_BUFFER memory, one or more of the four available sequence start address pointer needs to be re-programmed and updated during sensor runtime. The SEQ_BUFFER memory is mapped to the address space `0x3000` to `0x33FF`.

4.4.2 Clock configuration

As the device supports 38.4 MHz or 40 MHz crystals and 76.8 MHz or 80 MHz crystals to generate the required system clock frequency (76.8 MHz or 80 MHz), a total of seven registers must be configured accordingly, as shown in Table 8 and Table 9.

In case a 76.8 MHz or 80 MHz crystal or clock is used, there is also an option to run the PLL in a high-performance mode (frequency doubler only doubles the frequency of PLL_REF_CLK) to achieve improved phase-noise (PN) behavior. This is because the sigma-delta modulator (SDM) is clocked at twice the rate via CLK_FD.MOD.

Table 8 CLK_FD register configuration

XTAL/CLOCK	38.4 MHz	40 MHz	76.8 MHz	80 MHz
CLK_FD.MODE	1 (76.8 MHz PLL) 2 (76.8 MHz PLL)	1 (80 MHz PLL) 2 (80 MHz PLL)	0 (76.8 MHz PLL) 1 (153.6 MHz PLL) 2 (153.6 MHz PLL)	0 (80 MHz PLL) 1 (160 MHz PLL) 2 (160 MHz PLL)
CLK_FD.SRC_FREQ_SEL	0	0	1	1

Table 9 PLL register configuration

PLL_REF_CLK	76.8 MHz	80 MHz	153.6 MHz	160 MHz
PLL_CTRL0.FREQ_SEL	0	0	1	1
PLL_CTRL2.SWAL_SEL	28	28	16	16
PLL_CTRL3.FBDIV	42	40	19	18
SH_IDX#_START.FREQ	$2^{20}*((f_{PLL}/(4*f_{PLL_REF_CLK})) - 4*(PLL_CTRL3.FBDIV+3))$			
SH_IDX#_RAMP.STEP	$2^{20}*(\Delta f_{PLL}/(4*f_{PLL_REF_CLK}))$			

4.5 Quick start

The quick start will enable you to communicate with the device via the SPI to be capable of configuring the device properly. For a more detailed description of each module, check the respective chapter.

- **Reset:** after all power supplies are settled and the hardware reset (HW_RST) is released either via power-on reset (PoR) or the RST_N pin, the device can be configured via the Serial Peripheral Interface (SPI)
- **Bootloader:** in case the hardware bootloader is not required but cannot be disabled by setting the pins DIO2 and DIO3 to high, there is an option to disable the bootloader after the hardware reset is released. To do this, the software can perform an SPI write zero command with the address: `0x50` and data: `0x00000000` within 20 µs after e.g. the RST_N pin is set to high
- **SPI:** the device provides a Serial Peripheral Interface (SPI) slave that operates with clock polarity (CPOL) and clock phase (CPH) both set to zero, commonly referred to as mode 0. Initially, single SPI is selected with MOSI at pin DIO0 and MISO at pin DIO1. By default, the device uses LSB first for COMMAND, ADDRESS, and DATA phase. For the exact protocol, check the Section Serial Peripheral Interface (SPI)
  - In case MSB first is required, there is an option to change from LSB first to MSB first by setting `SPI_CTRL.MSB_SEL` to 1_B. However, this command must be executed with LSB first

- **System clock:** initially, the device is configured for operation with an 80 MHz crystal (XTAL). If either a 40 MHz XTAL or an external oscillator is used, the following must be changed:
  - To change the frequency from 80 MHz to 40 MHz, the bitfield `CLK_FD.SRC_FREQ_SEL` must be set to 0_B. Other bitfields in the register `CLK_FD` do not need any modification
  - If an external clock is used, the XOSC module needs to be re-configured with `CLK_XOSC_CTRL0.MODE` = 3_D
  - In case an external 40 MHz oscillator clock is provided, the frequency must be adjusted and described
  - Finally, the system clock can be enabled via a power-up request by setting `CLK_WUC.PURQ` = 1_B
  - Afterwards the XOSC module requires 0.5 ms to start up the system clock properly
- **I/O voltage:** by default, the system expects an I/O voltage of 1.2 V on VDDIO. If the digital interface operates with 1.8 V (VDDIO connected to 1.8 V), then the bitfield `IO_CTRL.IO_LVL` must be set to 1_B

Afterwards, it should be possible to read and modify all registers of the device and write to the sequence buffer memory. As a first check, it is recommended to read out the hardware ID of the device by performing a standard SPI read command to the register ID_HW. The expected value is `0x00098185`.

5 Function block description

5.1 Identification (ID)

After the system clock is available, the hardware chip ID can be read via the register ID_HW. The following three fields are available and can be read by the software via SPI:

- RF (0x85) indicates the version of the analog RF front-end of the chip
- DIG (0x81) indicates the version of the digital part of the chip
- ROM (0x00) indicates the version of the applied ROM mask

5.2 General timer (GTMR)

The general timer module shown in Figure 29 is used to generate trigger pulses to start up the system (XOSC and power domain) and to trigger a sequence start. The module is clocked with the 80 kHz low power RC oscillator (RCOSC). To allow more complex scenarios, the device features up to 4 independent general timer modules (GTMR0-3) with different timer threshold values and different repetition factors.

With `GTMR#_CTRL.EN` the module can be enabled if required. When disabled, all counters of the timers are stopped to save power.

There are multiple options to start a timer module. The device offers the flexibility to start the timer by an external trigger (TRIG#), by another general timer module (GTMR#), by a motion detection event (MD), by any sequence done (frame execution is over) (SEQ#_DONE), or by software (SW). The timer start trigger source can be selected via `GTMR#_CTRL.SRC_SEL`.

Using `GTMR#_CTRL.TRIG`, you can start the timer via software by selecting `SRC_SEL = 15`$_D$ (SW) provided it is configured accordingly.

To stop a timer, there is an option via software by using `GTMR#_CTRL.STOP` or via a motion detection event (MD_ULP_EVENT, MD_RD_EVENT) by programming `GTMR#_CTRL.STOP_EVENT_SEL`.

Based on the application it could be required to start the timer once and repeat it several times to generate multiple output triggers for multiple sequence execution (frame update rate). In this case, the number of repetitions need to be configured via `GTMR#_CTRL.REP` whereas zero enables an endless operation.

In some scenarios, it may be required to start a timer again before it expires. In such cases, the timer reloads its timer value again when receiving a start trigger. This feature can be enabled with `GTMR#_CTRL.RESTART_EN`. If disabled, additional start triggers are ignored while a timer is running and the GTMR#_ERR is set in the interrupt status register.

Note: If `GTMR#_CTRL.RESTART_EN` is disabled, any additional timer start trigger while the timer is still running is ignored by the timer and results in a timer error (`IR_STAT.GTMR#_ERR`) which needs to be cleared via `IR_CLEAR` register accordingly. The timer itself is not stopped in case the error is set.

With `GTMR#_TIMER.VALUE`, the duration (number of RCOSC clock cycles) of the timer can be set. After the configured time is over, a trigger pulse is generated. The minimum value of the timer is 2_D.

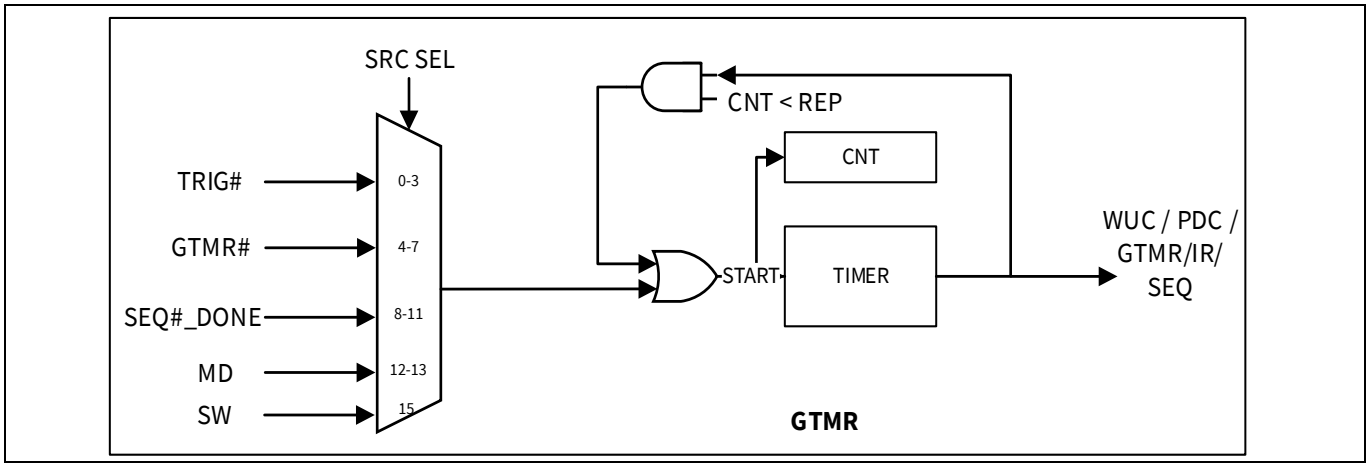


Figure 29 General timer module

Note: Ensure that the RC oscillator (RCOSC) is either in mode AUTO or ON to provide the RCOSC_CLK to the GTMR# module to be clocked properly.

5.3 Wakeup controller (WUC)

The wakeup controller module is used to automatically start up the crystal oscillator to provide a system clock (SYS_CLK) to the overall system. The SYS_CLK is required to power up the switched power domain, to clock the sequencer for executing configured sequences, to process data with the embedded signal processor (DSP) and to read and write registers as well as SRAMs.

Note: Ensure that the low power RCOSC is enabled via `CLK_RCOSC_CTRL.MODE` to clock the WUC module. CLK_* registers can be read and written without the SYS_CLK.

The wakeup routine can be triggered by one of the following events to change the state from SLEEP/DEEPSLEEP to ACTIVE:

- Power-up request (PURQ) via software to provide a system clock (`CLK_WUC.PURQ`)
- External trigger received via TRIG# to start a sequence if enabled (`SEQ#_CTRL.TRIG#_EN`)
- Elapsed time to start a sequence (`SEQ#_CTRL.GMTR#_EN`)
- Sequence started by software (`CLK_WUC.SEQ#_TRIG`)

A power-down transition can be triggered by one of the following events:

- Power-down request via software (`CLK_WUC.PDRQ`)
- Sequence and motion detection processing is done (`CLK_WUC.SEQ#_DONE = SLEEP or DEEPSLEEP`)

However, if the option to enter `SLEEP` or `DEEPSLEEP` is selected by hardware, the host must be aware that the device itself will disable the system clock (SYS_CLK) once the sequence or motion detection processing is over.

Therefore, it is up to the host to enable the XOSC first via sending a wakeup request (PURQ) before further SPI transactions are performed.

Note: If a sequence or post-processing is finished while there is an active SPI transaction ongoing, the system clock is not disabled but the WUC stays in the active state as long as the SPI transaction is ongoing to prevent the corruption of the SPI transaction.

The WUC can be in one of the three states as shown in Figure 30.

- **DEEPSLEEP:** the crystal oscillator is completely powered down (least power consumption) and requires ~350 µs to 500 µs to provide a system clock
- **SLEEP:** the crystal oscillator core is active with disabled shaper to save power. In this state, the system clock can recover within 0.1 µs
- **ACTIVE:** the crystal oscillator is active and provides the system clock (highest power consumption)

The power state that shall be applied after a sequence is over can be defined by `CLK_WUC.SEQ#_DONE`.

- **0_D:** DEEPSLEEP (if this setting is used, any SPI access to non CLK_* registers require a PURQ first)
- **1_D:** SLEEP
- **2_D:** ACTIVE
- **3_D:** RSVD (reserved)

Note: *In case motion detection is enabled, the transition to SLEEP or DEEPSLEEP is* delayed until motion detection processing is over.

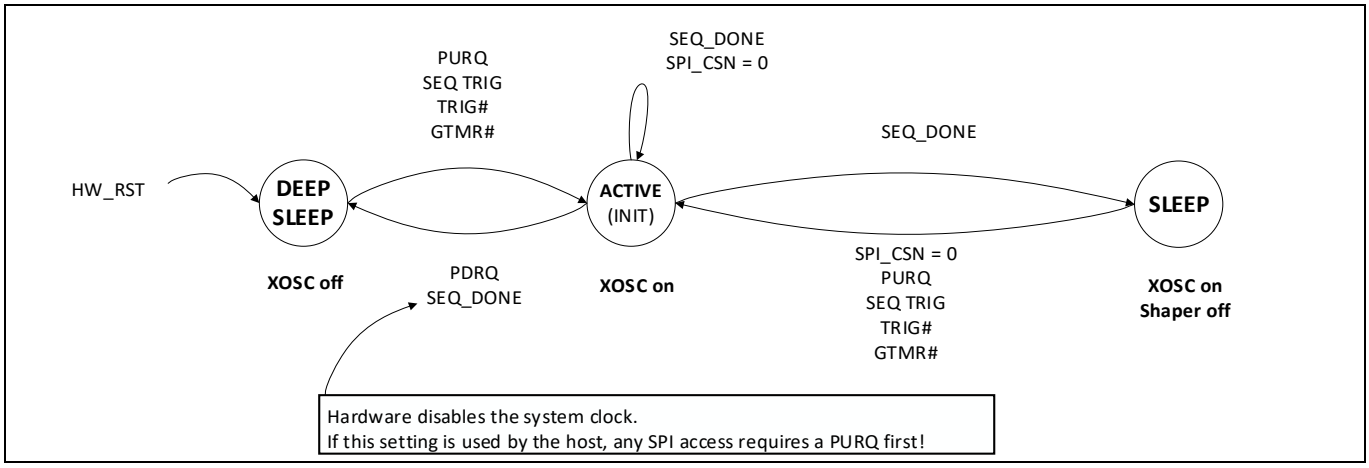


Figure 30 Wakeup controller states

5.4 Power domain controller (PDC)

Most of the functions in the digital domain are only required while the chip is executing a frame defined by a sequence or performing data processing, such as for motion detection. To save power during the inactive time, the device comes with a switchable power domain containing all blocks that are not required within this inactive time.

The power domain controller (PDC) manages the power-up and power-down of the switchable power domain (PD) based on operational requirements. Its embedded finite state machine (FSM) ensures proper application and release of power isolation, reset, and clock signals, as well as the correct switching timing required by the power switches. This routine takes approximately 3 µs, resulting in a latency between sequence trigger and sequence execution which is inevitable. The switching time can be adjusted via `PDC_CTRL.PWR_SWITCH_DEL` (typ. 2 µs) and `PDC_CTRL.PWR_UP_DEL` (typ. 1 µs). In case this latency is not acceptable for the application use-case, the device offers flexibility to keep the power domain always enabled by setting `PDC_CTRL.PM = 1D`. In this case, the PDC no longer dynamically powers up and down the switchable power domain. However, this comes with the drawback of increased power consumption in the digital domain during inactive periods (DEEPSLEEP).

Note: The state of the PDC can be read via `PDC_STAT.STATE` if the system clock (provided via XOSC) is active.

The following modules are part of the switchable power domain:

- Sequencer (SEQ)
- Finite state machine (FSM)
- Linear feedback shift register (LFSR)
- Average module for chirps (AVG)
- Presence detection processing (PD)

5.5 Interrupt router (IR)

The device provides the option to notify the host about certain events inside the chip via interrupts. For this, an interrupt router (IR), as shown in Figure 31 is implemented with one permanent status bit and one clear bit per event. To get additional flexibility, two individual enable masks are provided to create two separate interrupt events (IRQ0, IRQ1). Each interrupt is a logical OR combination of multiple, free selectable sources and can be configured via `IR0_EN.EN#` and `IR1_EN.EN#`. Sources that are not enabled are gated and are not contributing to interrupt events. The resulting interrupt events IRQ0 and IRQ1 can be flexibly mapped to any of the available digital input output pins (DIO) via `IO_DIO#.MAP`.

The typical use cases are:

- The device runs in standalone motion detection (MD) mode and notifies the host about a detected motion
- The execution of a sequence is over
- A selected sensor value is outside of the user defined limits (such as temperature)
- A certain fill level of the first-in first-out (FIFO) is reached during data acquisition
- A chip internal error has occurred

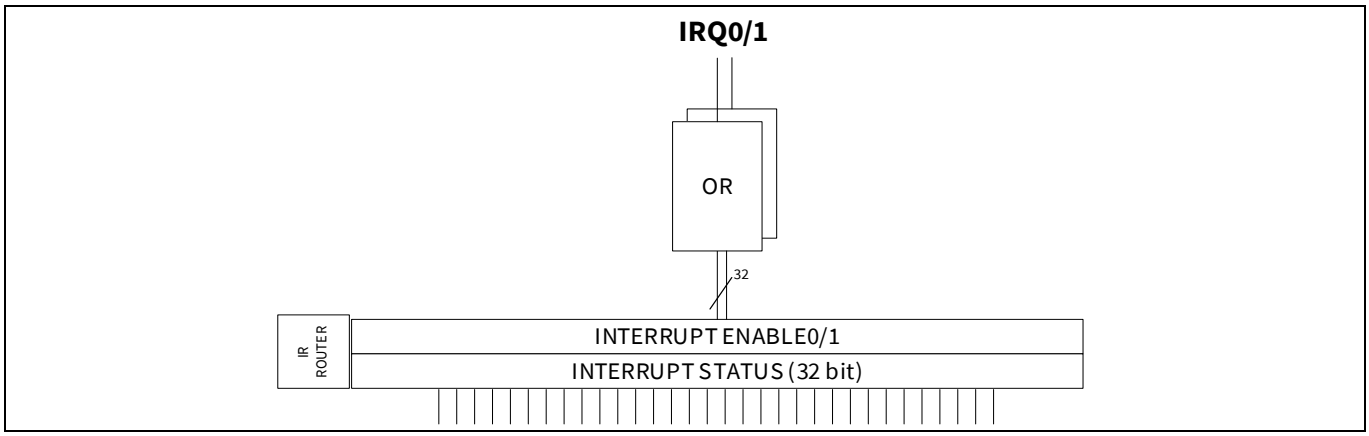


Figure 31 Interrupt router

Once an event occurs, it is permanently registered in the `IR_STAT.STAT#` field and must be cleared via the `IR_CLR.CLR#` field. However, if an interrupt status bit is cleared while its source remains active, the interrupt will be generated again.

For motion detection interrupt events (IR_STAT[4] (MD_ULP_EVENT), IR_STAT[5] (MD_RD_EVENT)), there is an option to automatically clear their status bits by one of the available embedded general timer modules (GTMR#). This feature can be enabled via `IR_CTRL.GTMR_MD_CLR_EN` and `IR_CTRL.MD_CLR_SRC_SEL`. For applications which require a presence flag, this can be used to clear the motion detection interrupt after specific time, as shown in Figure 32, such as 1 second after the last motion event.

Note: Ensure that the low-power RC oscillator (RCOSC) is enabled if the general timer module (GTMR) is required to automatically clear the motion detection event status.

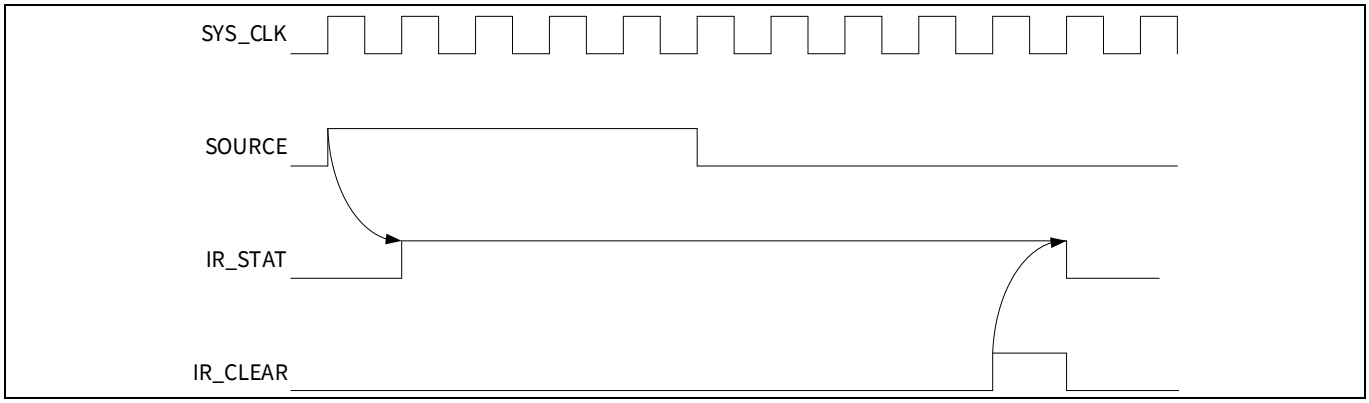


Figure 32 Interrupt router status

To be compatible with a wide range of microcontrollers and application processors, the device offers the capability to generate either LEVEL (0_B) or PULSE (1_B) interrupts. It can be flexibly configured for each interrupt event via `IR_CTRL.IR0_MODE` and `IR_CTRL.IR1_MODE`. In case the pulse mode is selected, there is also an option to extend the pulse to multiple clock cycles by configuring the respective DIO pin via `IO_DIO#.PULSE_LEN`. An example of pulse interrupt with a length of 4 clock cycles is shown in Figure 33. Interrupt router mapping table is shown in Table 10.

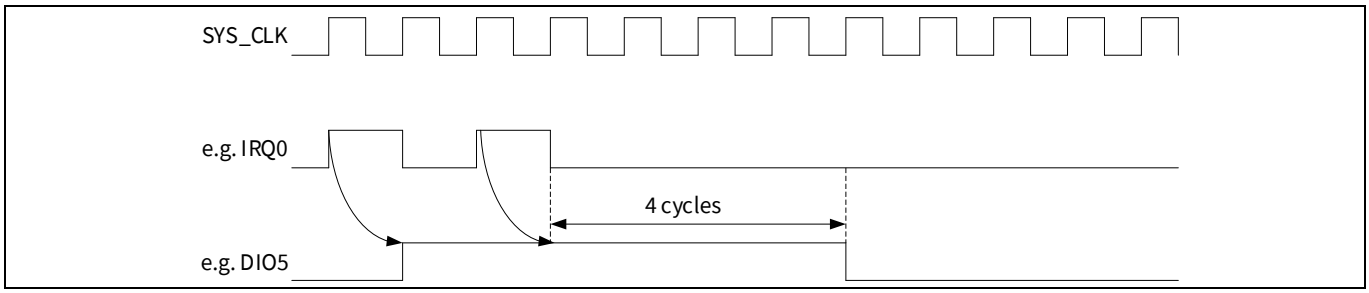


Figure 33 Example of pulse interrupt with a length of 4 clock cycles

Table 10 Interrupt router mapping table

Bit	Interrupt name	Description
0_D	SENSE0_MIN	Sensor 0 value fell below configured threshold
1_D	SENSE0_MAX	Sensor 0 value exceeded configured threshold
2_D	SENSE1_MIN	Sensor 1 value fell below configured threshold
3_D	SENSE1_MAX	Sensor 1 value exceeded configured threshold
4_D	MD_ULP_EVENT	Motion detected with ULP
5_D	MD_RD_EVENT	Motion detected with RD
6_D	RSVD	Reserved
7_D	FIFO_THRESHOLD	FIFO threshold exceeded
8_D	SEQ0_DONE	Sequence 0 execution done
9_D	SEQ1_DONE	Sequence 1 execution done
10_D	SEQ2_DONE	Sequence 2 execution done
11_D	SEQ3_DONE	Sequence 3 execution done
12_D	GTMR0_ERR	General timer 0 was triggered while still running
13_D	GTMR1_ERR	General timer 1 was triggered while still running
14_D	GTMR2_ERR	General timer 2 was triggered while still running
15_D	GTMR3_ERR	General timer 3 was triggered while still running
16_D	FIFO_UF_ERR	FIFO underflow occurred
17_D	FIFO_OF_ERR	FIFO overflow occurred
18_D to 21_D	RSVD	Reserved
22_D	BUS_ADDR_ERR	Illegal bus address access
23_D	SEQ_ERR	Sequence error occurred
24_D	PLL_RANGE_ERR	PLL sigma delta modulator range error occurred
25_D	SADC_ERR	SADC conversion or calibration error occurred
26_D	ADC1_ERR	ADC1 conversion or calibration error occurred
27_D	ADC2_ERR	ADC2 conversion or calibration error occurred
28_D	ADC3_ERR	ADC3 conversion or calibration error occurred
29_D to 31_D	RSVD	Reserved

5.6 Phase lock loop (PLL)

The analog phase-locked loop (PLL) is designed to generate high-performance frequency chirps in the range of 57.0 GHz to 64.0 GHz. The modulation is performed inside the PLL bandwidth (in-band-modulation) using an analog charge-pump-based fractional-N RF architecture. Furthermore, it features a shape generator (RAMPER) with high flexibility to allow different shapes and duration times of the ramps. The loop requires a low noise reference clock (PLL_REF_CLK) with a nominal frequency of SYS_CLK or 2*SYS_CLK based on the frequency-doubler settings (`CLK_FD.MODE` and `CLK_FD_SRC_FREQ_SEL`).

The PLL is located in the analog domain and comes with its own embedded LDO's to generate all the required low noise power supplies. Therefore, it is required to start up the PLL before chirping and power it down when the chirp or frame is completed via sequencer commands. That ensures the lowest possible power consumption while the sensor is in DeepSleep state.

All required PLL configurations and shape-specific parameters (start frequency, ramp step) are stored in registers (`PLL_CTRL*`, `SH_IDX*`) inside the digital always-on domain to retain their content if VDDD is applied properly. The overall PLL status can be monitored via the status register PLL_STAT to get information about proper locking, ramp progress, and sigma delta modulation error (range error).

The RF frequency ramps generated by the PLL are controlled via the `SH_IDX*` registers to generate up-chirps (increasing frequency over time) and down-chirps (decreasing frequency over time). Each RF frequency ramp is defined by the start frequency programmed to `SH_IDX#_START.FREQ`, the ramp slope programmed to `SH_IDX#_RAMP.STEP`, and the ramp time programmed to `SH_IDX#_TR_RAMP.MUL`. The multiplier refers to the number of system clock cycles (SYS_CLK). The frequency increase per system clock cycle (ramp step) is shown in Figure 34.

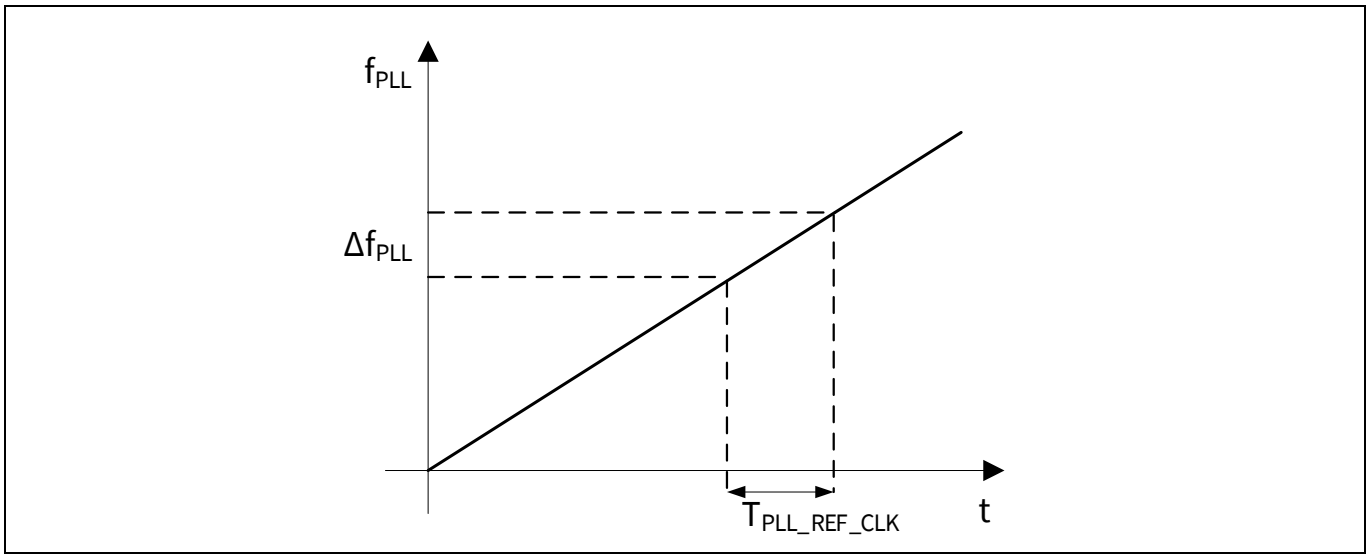


Figure 34 PLL ramp step per system clock cycle

5.6.1 PLL parameter calculation

The start frequency programmed to the register `SH_IDX#_START.FREQ`, is in the UQ5.20 representing 5 unsigned integer bits and 20 fractional bits. It can be calculated using Equation 1.

$$SH_{IDX}\#_{START}.FREQ = 2^{20} \times \left[\frac{f_{PLL_{START}}}{4 \times f_{PLL_{REF_{CLK}}}} - 4 \times (PLL_{CTRL3}.FBDIV + 3)\right]$$

Equation 1 PLL start frequency

The ramp step per system clock cycle, programmed to the register `SH_IDX#_RAMP.STEP`, is represented as a two's-complement value in SQ5.20 format, consisting of 1 sign bit, 4 integer bits, and 20 fractional bits. It can be calculated using Equation 2.

$$SH_{IDX}\#_{RAMP}.STEP = 2^{20} \times \frac{\Delta f_{PLL}}{4 \times f_{PLL_{REF_{CLK}}}}$$

Equation 2 PLL ramp frequency step per clock cycle

The ramp duration time programmed to the register `SH_IDX#_TR_RAMP.MUL`, is represented with 17 integer bits and refers to the system clock cycles (SYS_CLK). It can be calculated using Equation 3.

$$SH_{IDX}\#_{TR_{RAMP}}.MUL = ramp_{time[s]} \times f_{SYS_{CLK}}[Hz]$$

Equation 3 Ramp duration time multiplier

The resulting PLL bandwidth for the chirp can be calculated using the number of clock cycles per ramp (`SH_IDX#_TR_RAMP.MUL`) multiplied by the frequency increase per system clock cycle.

Note: The effective bandwidth of the chirp is different from the PLL bandwidth due to a delayed ADC sampling start and a hardware-controlled transmit disable. For a detailed description of chirp timing, see the Section "Chirp timing".

To be capable of operating the PLL with different input frequencies, the feedback divider and swallow value need to be configured based on the used PLL_REF_CLK frequency. Table 11 represents the required input values for these parameters.

Table 11 PLL parameter based on the used PLL_REF_CLK frequency

Parameter	76.8 MHz	80 MHz	153.6 MHz	160 MHz
PLL_CTRL2.SWAL_SEL	28_D	28_D	16_D	16_D
PLL_CTRL3.FBDIV	42_D	40_D	19_D	18_D

5.6.2 PLL timings

If required by the application, instead of up-chirps or down-chirps, sawtooth ramps or inverse-sawtooth ramps with fast fly back to the start frequency can also be programmed. In such cases, the fast-down timer can be flexibly programmed via the register `PLL_CTRL2.TR_FSTDN` referring to system clock cycles (SYS_CLK) and must be enabled via `SH_IDX#_CTRL.FSTDN_EN`.

Additionally, there is an option to delay the ramp start of the PLL or extend the chirp duration by additional PLL timings. These timings can be programmed via `SH_IDX#_TR_START.MUL` and `SH_IDX#_TR_END.MUL` and are referring to system clock cycles (SYS_CLK).

Considering all previously mentioned PLL timers, the overall PLL chirp timing can be calculated using Equation 4 including a constant number of cycles (offset). PLL timing is shown in Figure 35.

$$T_{Chirp} = TR_{START} + TR_{RAMP} + (TR_{FSTDN} + 1) + TR_{END} + 16$$

Equation 4 PLL chirp timing

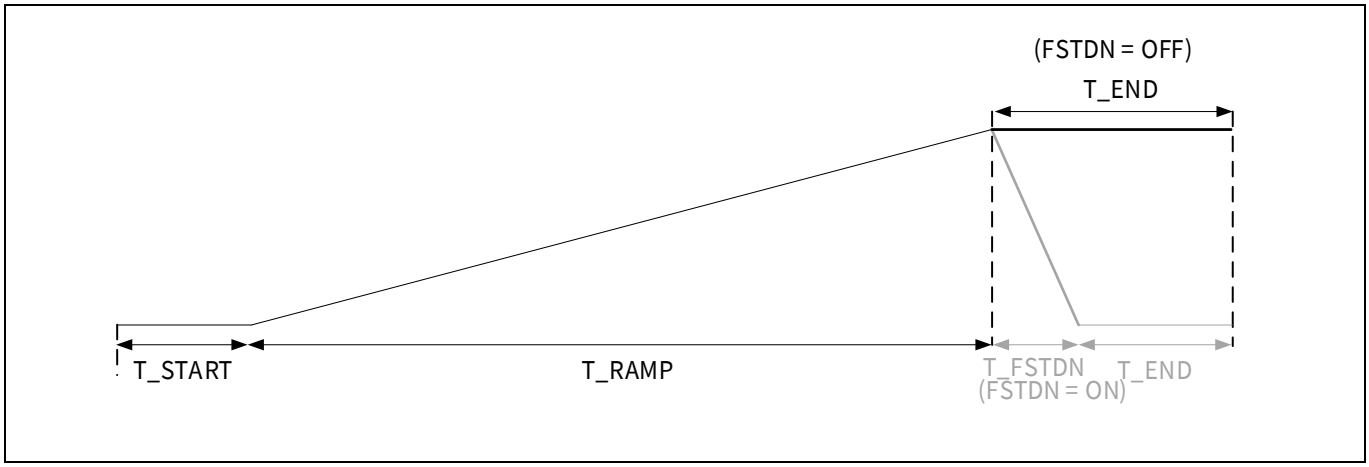


Figure 35 PLL timing

5.6.3 PLL configuration

To cover a wide range of application use-cases, the PLL provides the option to adjust its settings for the internal charge-pump current and its loop-filter resistor per shape configuration via the register `SH_IDX#_CTRL`. These settings have an impact on the PLL bandwidth and on the phase noise (PN).

Settings from Table 12 are recommended by Infineon.

Table 12 PLL control settings

Parameter (SH_IDX#_CTRL)	60 GHz to 61 GHz	57 GHz to 64 GHz
RLF1	0_D (1 kΩ)	TBD
RLF2	2_D (1 kΩ)	TBD
SDM_TYPE	2_D (MASH3)	TBD
CP_SEL	0_D (400 µA)	TBD

5.7 Radio frequency front-end (RFFE)

The RF front-end (RFFE) of this sensor is responsible for generating the 60 GHz signal, transmitting it via the TX antenna, receiving the reflected echo on the RX antenna and mixing it down to an intermediate frequency up to several MHz.

For this purpose, the RFFE is divided into two parts: the transmitting (TX) part, which consists of a binary phase-shift keying (BPSK) feature and a programmable power amplifier (PA), and the receiving (RX) part, which consists of three channels, each consisting of a passive mixer, a transimpedance amplifier (TIA), and a phase shifter (PS) with a range of up to 180°.

Furthermore, several built-in test structures (BITE) are included, providing test capabilities for production test and end of line (EOL) testing.

5.7.1 Transmitter (TX)

The RFFE transmit channel shown in Figure 36 supports a programmable power amplifier to adjust the transmitted power for system optimization and the binary phase-shift keying per chirp for interference mitigation.

The two-stage power amplifier can be controlled via the registers `CS_IDX#_TX1.STG1_CTRL` and `CS_IDX#_TX1.STG2_CTRL`, each 8-bit wide. To keep the transmission path off during the settling time of the PLL at the beginning of a chirp or to prevent out of band transmissions at the end of a chirp, the PA can be enabled or disabled at any point in time. The TX enable delay can be configured on a per-shape basis and must be programmed via the register `SH_IDX#_TR_TXEN`.

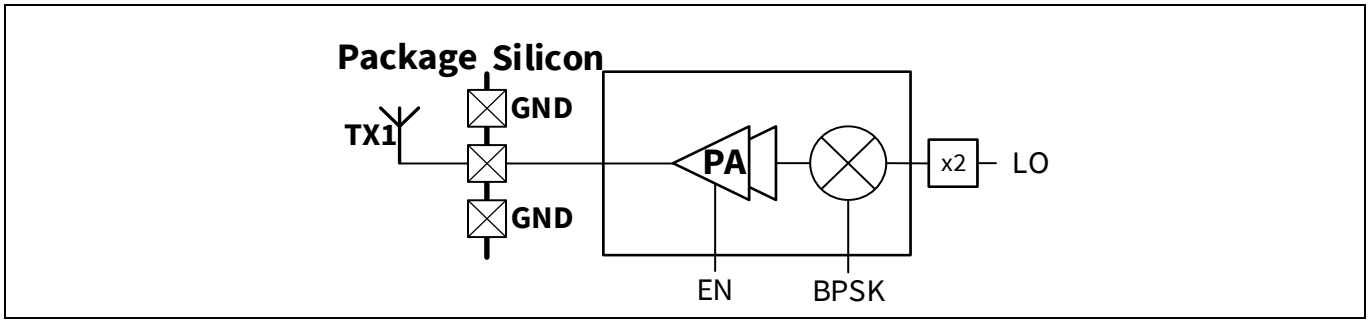


Figure 36 Transmitter channel (TX)

Recommended PA control values can be found on the Table 13.

Table 13 PA control settings

Parameter	Value
STG1_CTRL	16_D
STG2_CTRL	92_D (64_D to 127_D)

Figure 37 shows the conducted output power with respect to STG2 control settings.

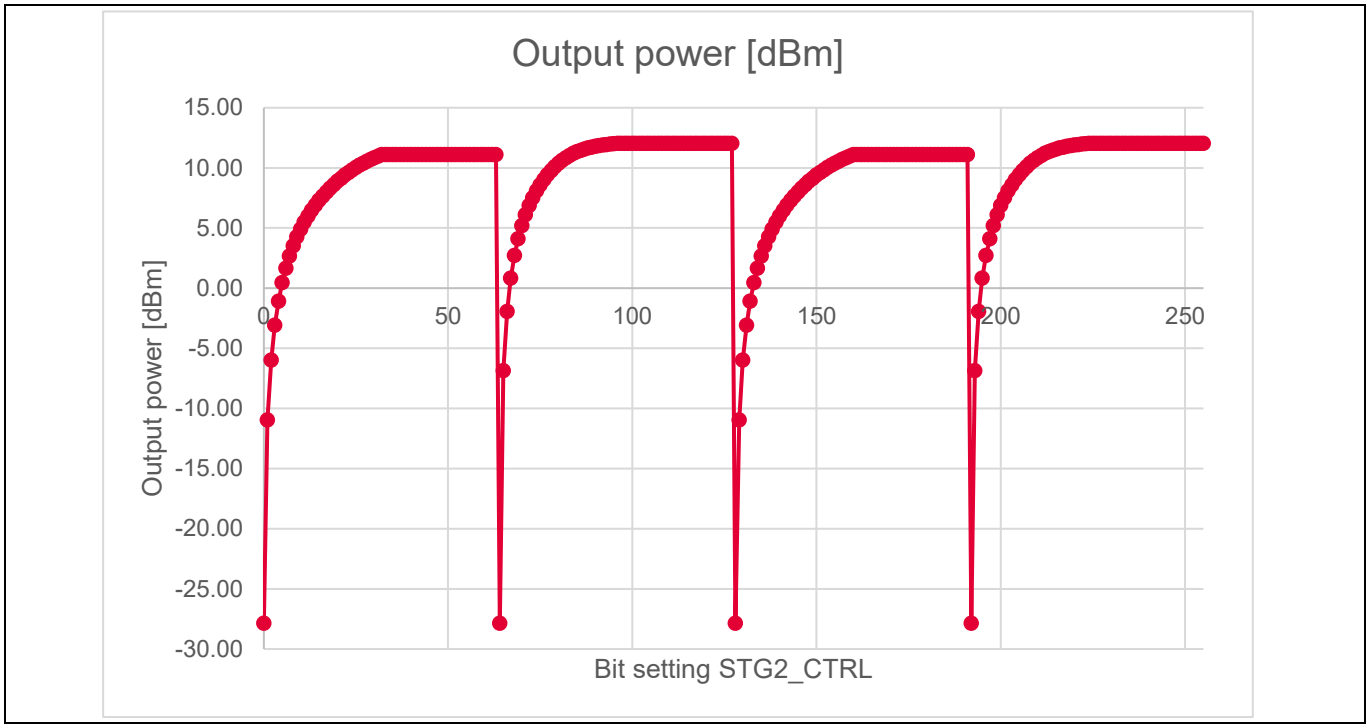


Figure 37 PA (STG2) control settings (conducted power)

The binary phase-shift keying is controlled via an embedded linear feedback shift register (LFSR) module. It provides the option to generate a 16-bits based m-sequence with flexible configurable length, polynomial (taps), and starting seed. The lowest bit (LSB) is mapped to the BPSK input to shift the phase of the transmitted signal by 180° when enabled or by 0° when disabled. The shift is performed on chirp basis and changes only between chirps.

5.7.2 Receiver (RX)

Each RFFE receive channel shown in Figure 38 features a passive mixer, a transimpedance amplifier (TIA), a phase shifter (PS) up to 180°, and built in test (BITE) to generate internal test signals.

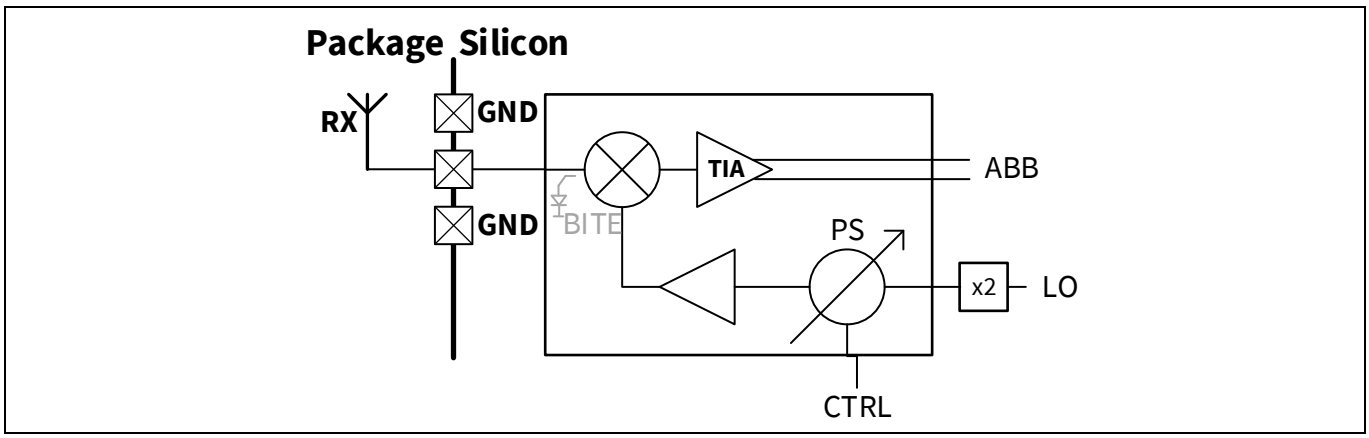


Figure 38 Receiver channel

All receive channels can be individually enabled or disabled via the register CS_IDX#_RX#.RX_EN based on the system requirements. The transimpedance amplifier (TIA) bias current can be separately controlled via `CS_IDX#_RX#.TIA_CTRL`. However, it is not recommended to change its default value 10_D.

The embedded phase shifter (PS) for each RX channel can be used to optimize DC leakage, such as that generated by close targets (For example, a radome). To cover 180° of a 60 GHz signal, the following settings from Table 14 should be applied to the register `CS_IDX#_RX#.PHS_CTRL`.

Table 14 PS control settings

Parameter	Value
PHS_CTRL	0_D (0_D to 42_D)

5.7.3 Analog baseband (ABB)

The analog baseband (ABB) filters (Figure 39) and amplifiers adjust the IF signals after RX mixer to fulfill the system requirements. The signal levels are controlled in a way to drive the ADC inputs in full scale without clipping.

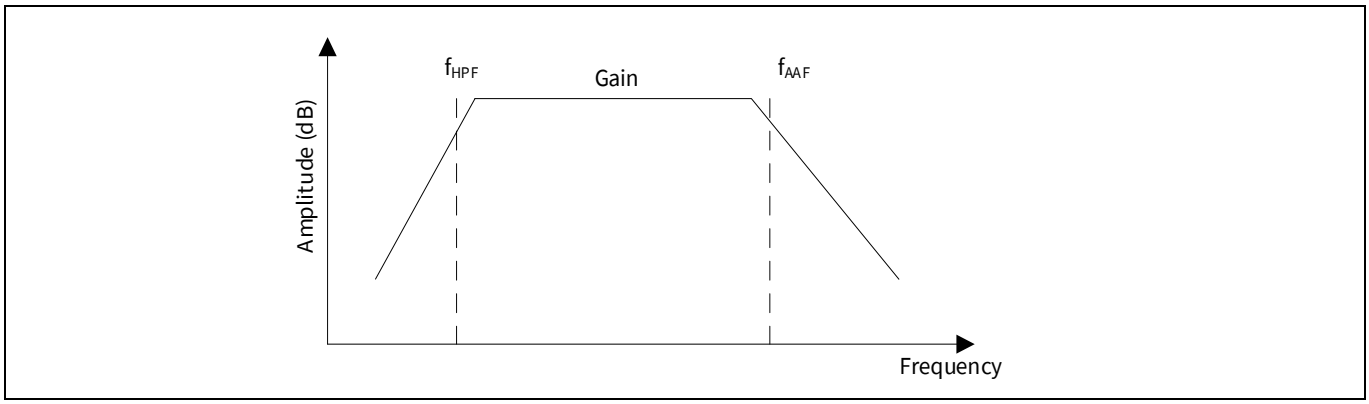


Figure 39 Analog baseband filter characteristic

There is one ABB block per receiver (RX) channel. Each channel consists of a high-pass filter (HPF), a variable gain amplifier (VGA), an anti-aliasing filter (AAF), and a driver for the ADC shown, as in Figure 40.

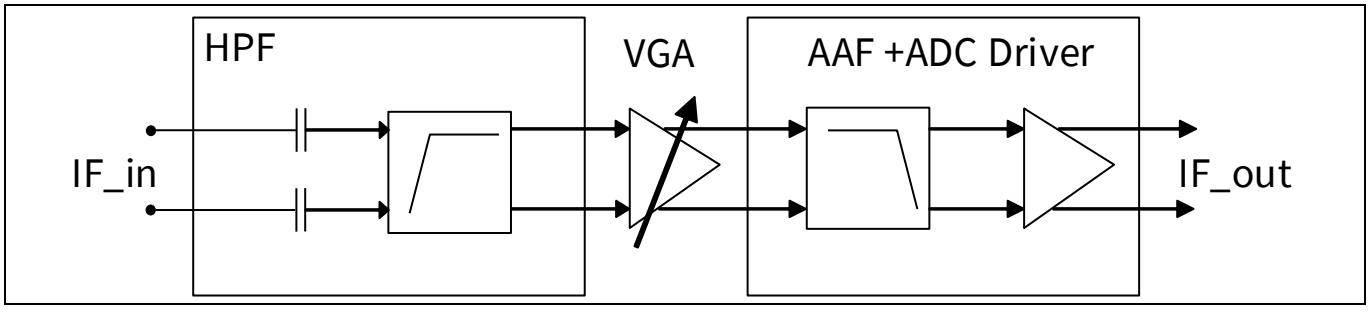


Figure 40 Analog baseband block diagram

5.7.3.1 High-pass filter (HPF)

The HPF is used to remove the DC offset at the output of the RX mixer and suppress the reflected signal from close unwanted targets (such as a radome).

The HPF can be tuned to accommodate different cut-off frequencies f_{HPF} according to different modulation parameters. Given the expected power levels the radar system will encounter, the HPF does not degrade the linearity of the system. As shown in Table 15, the HPF of each RX channel can be set to one of the four available settings via `CS_IDX#_RX#.ABB_HPF1_SEL`.

Table 15 High-pass filter settings

Setting	Description
0_D	100 kHz
1_D	350 kHz
2_D	650 kHz
3_D	1000 kHz

The high-pass filter can be bypassed to reduce the corner frequency to 1 kHz. If required by the application, the field `CS_IDX#_RX#.ABB_HPF2_BYP_EN` must be set.

5.7.3.2 Variable gain amplifier (VGA)

After the high-pass filter, the IF signals are amplified by the amplifier stage. The selectable voltage gain can be adjusted in the VGA in 13 steps of 3 dB each, up to a maximum gain of 60 dB. The VGA is followed by a three pole anti-aliasing filter. All possible configuration settings are listed in Table 16.

Table 16 Variable gain amplifier settings

Setting	Description
0_D	24 dB
1_D	27 dB
2_D	30 dB
3_D	33 dB
4_D	36 dB
5_D	39 dB
6_D	42 dB
7_D	45 dB
8_D	48 dB
9_D	51 dB
10_D	54 dB
11_D	57 dB
12_D	60 dB
13_D to 15_D	RSVD

5.7.3.3 Anti-aliasing filter (AAF)

After the variable gain amplifier, the IF signals are limited by an anti-aliasing filter (AAF), as shown in Figure 41, to avoid distortion effects (aliasing) by fulfilling the Nyquist criterion $f_{ADC} > 2 \times f_{max}$. As shown in Table 17, the AAF of each RX channel can be set to one of the four available settings via `CS_IDX#_RX#.ABB_AAF_SEL`.

Table 17 Anti-aliasing filter settings

Setting	Description
0_D	1 MHz
1_D	2 MHz
2_D	5 MHz
3_D	6 MHz

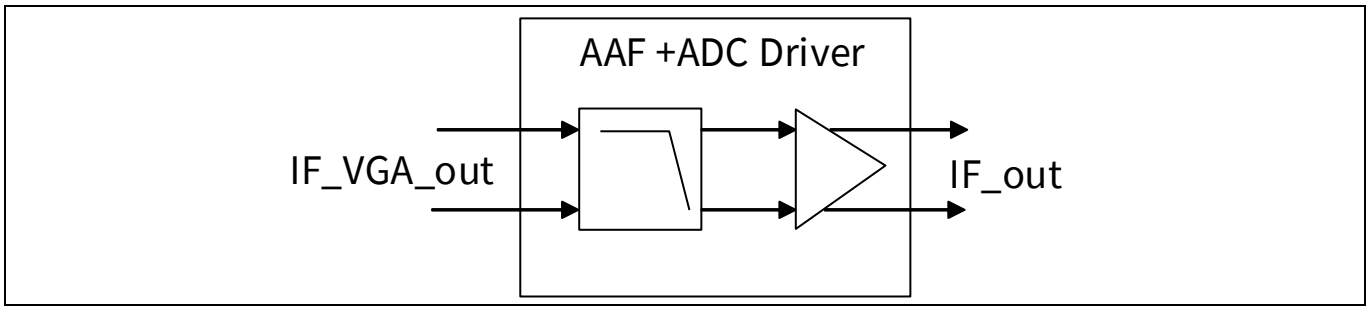


Figure 41 Anti-aliasing filter block diagram

5.7.4 Analog digital converter (ADC)

The multi-ADC (MADC) block shown in Figure 42 consists of four differential successive approximation register (SAR) ADCs. All are sharing the same common analog (CA) module, providing reference voltage and reference current to them. There is one ADC per receive (RX) channel and one dedicated ADC to sense available on-chip sensors like power-detector or temperature. Each RX ADC captures the differential intermediate frequency (IF) output signal of the analog baseband (ABB) and converts it into a digital representation of the same. The internal voltage is generated by a separate low dropout voltage regulator (LDO) supplied via VDDA, whereas the digital logic is supplied by the digital LDO.

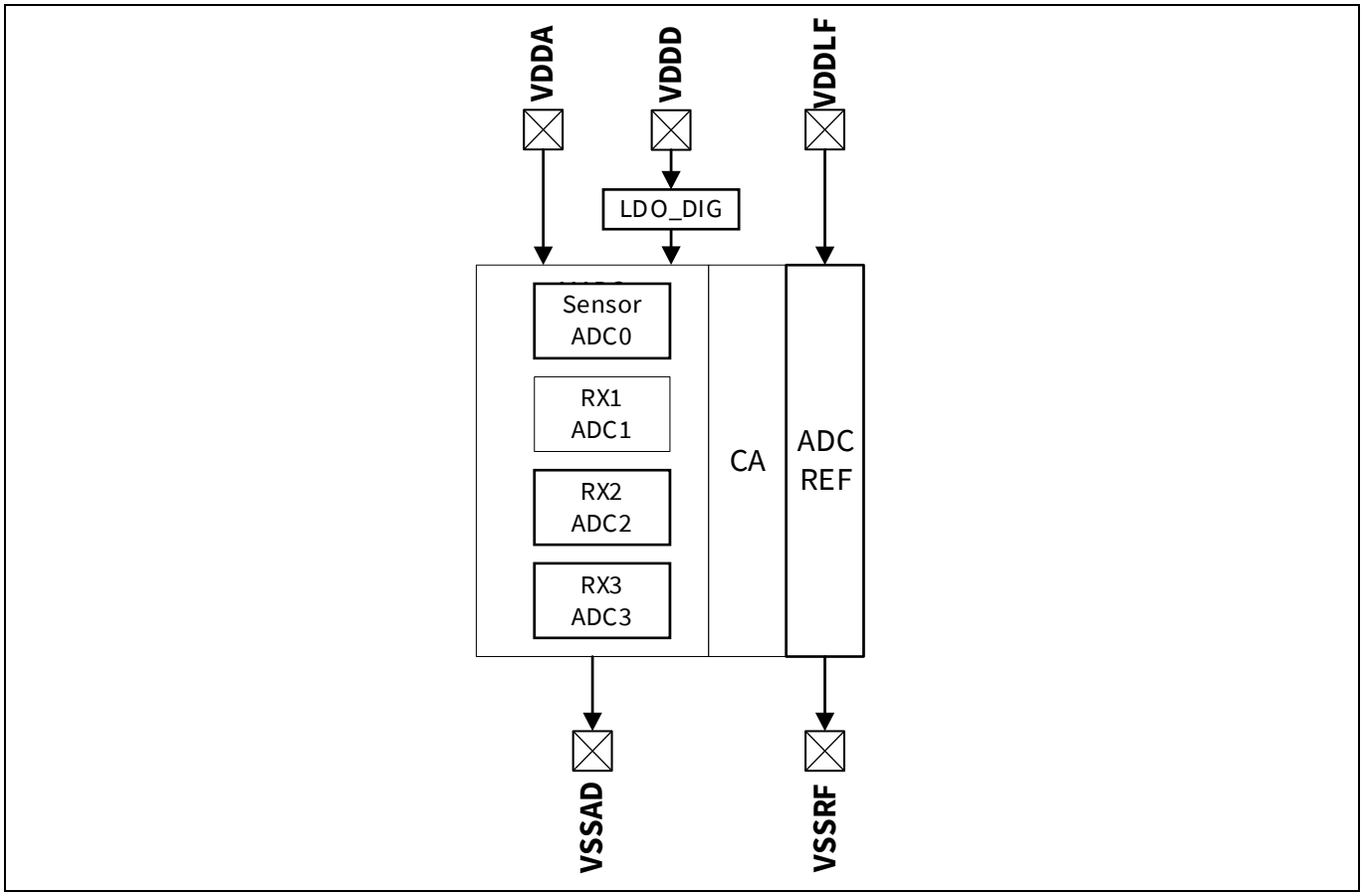


Figure 42 Block diagram of multi ADC

The ADC supports a differential input voltage of ±1.08 V which is converted to a digital representation with 12-bits and 4 fractional bits. To benefit from the additional four fractional bits an ADC calibration needs to be performed. This is typically done during device startup by executing the special ADC_CAL sequence. After the special sequence was executed, the calibration results are read from ADC and stored in registers. During each ADC startup within a sequence, these values are applied to the ADC so that the calibration phase of the ADC can be skipped.

The ADC is clocked by the system clock (SYS_CLK) and requires four clock cycles for a single conversion. The first clock cycle is used for capturing the voltage, whereas the leftover three clock cycles are used for the conversion. With a system frequency of 80 MHz, the maximum sample rate of the ADCs is 20 MS per second (80 MHz divided by 4). There is an option to configure the sample rate divider via the register `ADC_IDX#_CTRL.SR_DIV` to the required value. As sample rates higher than 20 MSps cannot be achieved, the minimum divider value is 4_D.

5.7.5 Phase shifter (PHS)

The device features a phase shifter (PHS), per receiver channel to shift the LO signal between 0° and 180°, as shown in Figure 43. For that purpose, a 6-bit control word is applied and can be adjusted via the channel set index settings (CS_IDX) for different applications. The PHS can be used to reduce the direct current (DC) part of each RX channel so that the DC offset is minimized. For that purpose, each RX PHS needs to be programmed with its own optimized PHS control word.

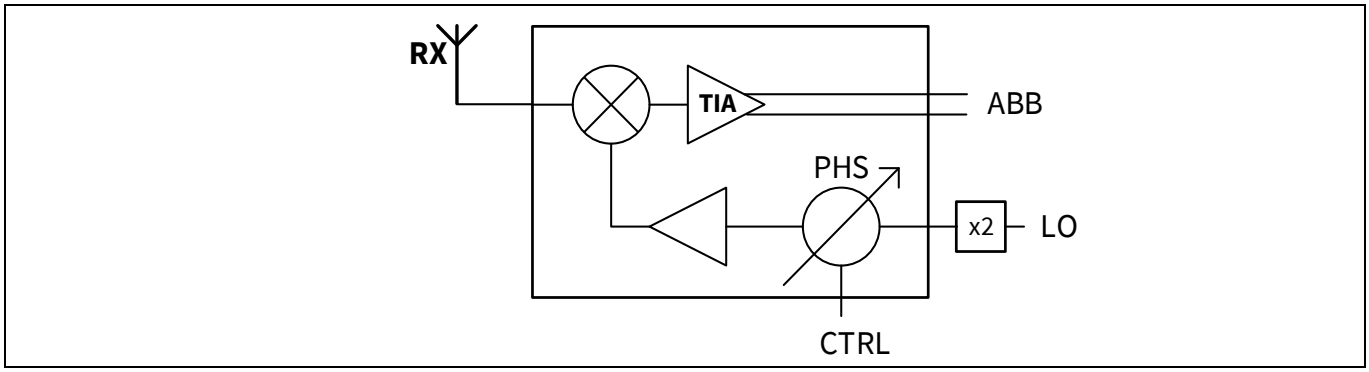


Figure 43 RX phase shifter

5.8 Sensor ADC (SADC)

The integrated sensor ADC can be used to measure the system clock duty-cycle, the on-chip temperature and the power on transmit (TX) and receive (RX) channel.

Like the ADC for the receiver channels, there are two sensor ADC configuration sets available (SADC_IDX0 and SADC_IDX1) to use two different settings. In addition to the sampling rate (`SADC_IDX#_CTRL.SR_DIV`) and the decimation factor (`SADC_IDX#_CTRL.DF`), the sensors listed in Table 18 can be selected using (`SADC_IDX#_CTRL.SEL`).

Table 18 Sensor ADC selection

Setting	Name	Description
0_D	SYSCLK_DC	Duty cycle of system clock after FD
1_D	TEMP	Temperature sensor
2_D	PD_RX1	Power detector sensor RX1
3_D	PD_RX2	Power detector sensor RX2
4_D	PD_RX3	Power detector sensor RX3
5_D to 7_D	RSVD	Reserved
8_D	PD_TX1_FWD	Power detector sensor TX1 (forward)
9_D	PD_TX1_REF	Power detector sensor TX1 (reflected)
10_D to 31_D	RSVD	Reserved

In normal operation, the sensor ADC is triggered during a CHIRP command to measure the sensor which is selected in the chosen SADC index register (SADC_IDX).

Note: As there is physically only one sensor ADC, it is not possible to measure more than one sensor at the same time. If more than one sensor needs to be measured, one CHIRP command per sensor readout is required. Prior to a sensor measurement, ADC calibration needs to be performed.

5.8.1 SADC result threshold

This feature allows tracking the results of two sensors during normal operation and notifying the microcontroller or application processor when the measured values exceed the normal operational ranges. Both minimum (SADC_IDX#_TH_MIN.THRESHOLD) and maximum (SADC_IDX#_TH_MAX.THRESHOLD) threshold values are supported. By default, its threshold values are set to minimum and maximum ADC values accordingly. The bitfield SADC_IDX#_STAT.MAX is set when the sensor result is above the maximum threshold and cleared when below. Similar to that, the bitfield SADC_IDX#_STAT.MIN is set when the sensor result is below the minimum threshold and cleared when above. This is shown in Figure 44.

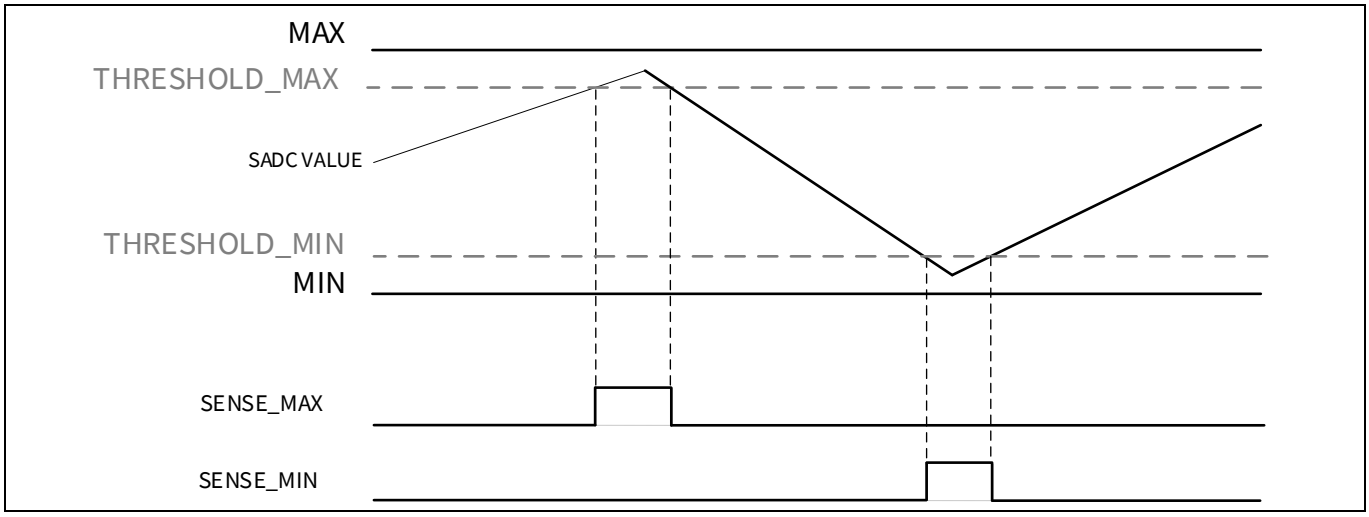


Figure 44 Sensor ADC result thresholds

5.9 Linear feedback shift register (LFSR)

The device features two embedded linear feedback shift registers (LFSR) with configurable length up to 16-bit (LFSR#_CTRL.LENGTH), configurable polynomials (LFSR#_POLY.POLYNOM) and configurable starting seeds (LFSR#_SEED.SEED). Both can be individually enabled via LFSR#_CTRL.EN and optionally be inverted (LFSR#_CTRL.INV) if required. LFSR block diagram is shown in Figure 45.

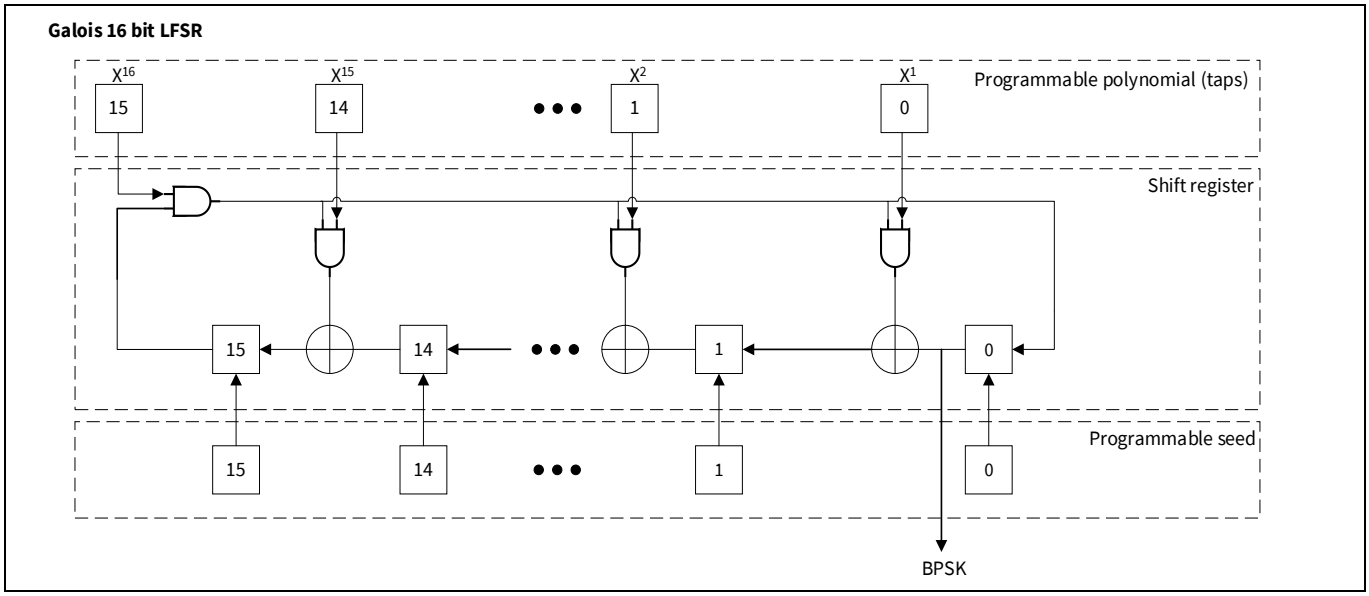


Figure 45 LFSR block diagram

The LFSR can be used in one of the following scenarios:

- Binary phase-shift keying (BPSK) pattern generation
- Test data generation to replace ADC data

5.9.1 Binary phase-shift keying (BPSK) pattern generation

There is an option to use binary phase-shift keying (BPSK) as modulation on transmitter side for interference mitigation. To be fully flexible regarding length, polynomial, and starting seed, two linear feedback shift registers (LFSR) are implemented.

Each LFSR can be used to define a maximum sequence called m-sequence to generate a unique bit pattern. The LSB of this pattern can be used to apply BPSK modulation while chirping during a frame. To ensure the same transmitted phase (0° or 180°) during a chirp, the LFSR shift operation is performed once at the beginning of a chirp before the power amplifier (PA) on the transmit (TX) path is enabled. Once the sequence is done, the LFSR is reset to its default state. LFSR configuration options are provided in Table 19.

Table 19 LFSR configuration options

Bits	Sequence length	Taps	LFSR#_POLY.POLYNOM	LFSR#_CTRL.LENGTH
2	3	2, 1	0x0003	0x1 (LEN2)
3	7	3, 2	0x0006	0x2 (LEN3)
4	15	4, 3	0x000C	0x3 (LEN4)
5	31	5, 4, 3, 2	0x001E	0x4 (LEN5)
6	63	6, 5, 3, 2	0x0036	0x5 (LEN6)
7	127	7, 6, 5, 4	0x0078	0x6 (LEN7)
8	255	8, 6, 5, 4	0x00B8	0x7 (LEN8)
9	511	9, 8, 6, 5	0x01B0	0x8 (LEN9)
10	1023	10, 9, 7, 6	0x0360	0x9 (LEN10)
11	2047	11, 10, 9, 7	0x0740	0xA (LEN11)
12	4095	12, 11, 8, 6	0x0CA0	0xB (LEN12)
13	8191	13, 12, 10, 9	0x1B00	0xC (LEN13)
14	16383	14, 13, 11, 9	0x3500	0xD (LEN14)
15	32767	15, 14, 13, 11	0x7400	0xE (LEN15)
16	65535	16, 14, 13, 11	0xB400	0xF (LEN16)

A configuration with LENGTH=4_D, POLYNOMIAL=0x0009 and SEED=0x1 will generate the following pattern:

Code listing 6 LFSR example

```
0001, 0010, 0100, 1000, 0011, 0110, 1100, 1011, 0101, 1010, …
```

5.9.2 Test data generation

For test purposes, there is the option to use one or both embedded LFSR to generate known pseudo random numbers to test the data streaming path from FIFO via the SPI to the host microcontroller. In such cases, each ADC conversion shifts the LFSR by one pattern and the unknown ADC output value is replaced by the known LFSR output data.

To match the ADC output bit width of 16-bits, the length of the selected LFSR needs to be configured to 16 (0xF: LEN16). Once the LFSR configuration is completed, the ADC data replacement must be configured via `LFSR_TEST.EN`. Any triggered sequence that generates data during chirping is now generating known data which is stored in the FIFO.

The available ADC data replacement options are shown in Table 20.

Table 20 LFSR to ADC mapping options

`LFSR_TEST.EN`	ADC	Comment
0	Disabled	ADC output data is not replaced
1	ADC1	ADC1 output is replaced by LFSR1
2	ADC2	ADC2 output is replaced by LFSR2
3	ADC1 & ADC2	ADC1 and ADC2 out data is replaced by LFSR1 and LFSR2, respectively

Figure 46 shows the LFSR test data flow.

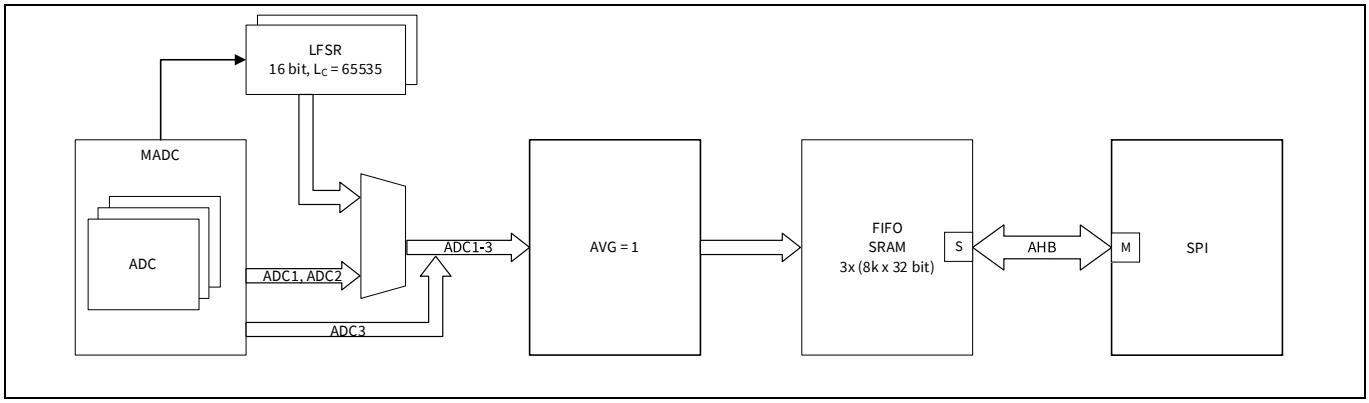


Figure 46 LFSR test data flow

6 Functional description

The XENSIV™ BGT60CUTR13AIP radar sensor is designed to operate with the lowest possible power consumption. To achieve this, the chip includes two oscillators (OSC), several domain specific low dropout (LDO) voltage regulators, power domains and switchable memories (SRAM).

To enhance the system's low power capabilities, the device features a fully integrated autonomous motion detection signal processing chain. This chain notifies the host when a motion is detected within the field of view (FoV) of the sensor. If there is no activity, the system processor can remain in its low power state and only power up when a motion event occurs to reduce the overall power consumption. This is especially beneficial for battery-powered applications, as it greatly extends the device's operational lifetime.

Most of the time, the sensor itself remains in its lowest power state, known as DeepSleep. In this state, only the digital LDO, the low power RC oscillator (RCOSC) and the general timer (GTMR) modules are active. The GTMR module allows the device to run with any update rate (such as, 1 Hz, 4 Hz, 10 Hz) and waking it up periodically.

Due to its embedded wakeup controller (WUC) unit and its power domain controller (PDC), no interaction with the sensor from system side is required to startup the high accurate crystal oscillator (XOSC) and the switchable power domain. Once the system clock (SYS_CLK) and the power domain is available, the frame defined by the sequence is executed automatically. The data is then analyzed by the embedded motion detection processing pipeline to identify ongoing motions.

The following chapters describe the modules required for frame execution, data processing, and storing in the memory as well as the motion detection processing pipeline in more detail.

6.1 Radar frame definition

In frequency-modulated continuous wave (FMCW) radar technology, a "chirp" refers to a specific signal waveform that is characterized by a linear change in frequency over time. This concept is foundational to how FMCW radars operate, enabling them to measure distances and velocities of targets effectively. The time interval between two consecutive FMCW radar chirps is called chirp repetition time, as shown in Figure 47. It is the time taken by the radar system to complete one cycle of frequency modulation and return to its initial state.

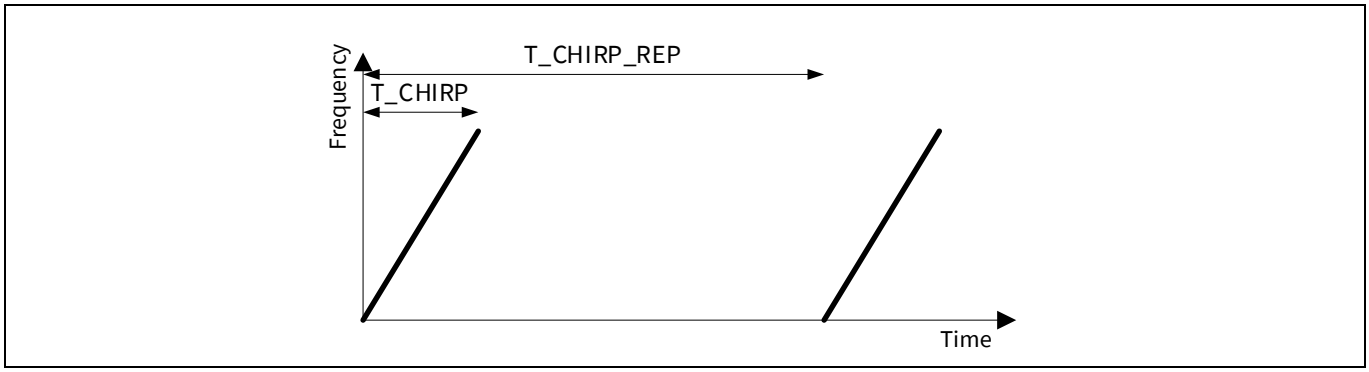


Figure 47 Chirp

In frequency modulated continuous wave (FMCW) radar technology, "bursts" refer to a series of transmitted chirp signals that are emitted in rapid succession over a defined time period (T_CHIRP_REP). The time interval between two consecutive bursts of FMCW radar chirps is called burst repetition time (T_BURST_REP). Total time taken to transmit N number of transmitted chirps is called observation time of a burst (T_BURST = N * T_CHIRP_REP), as shown in Figure 48.

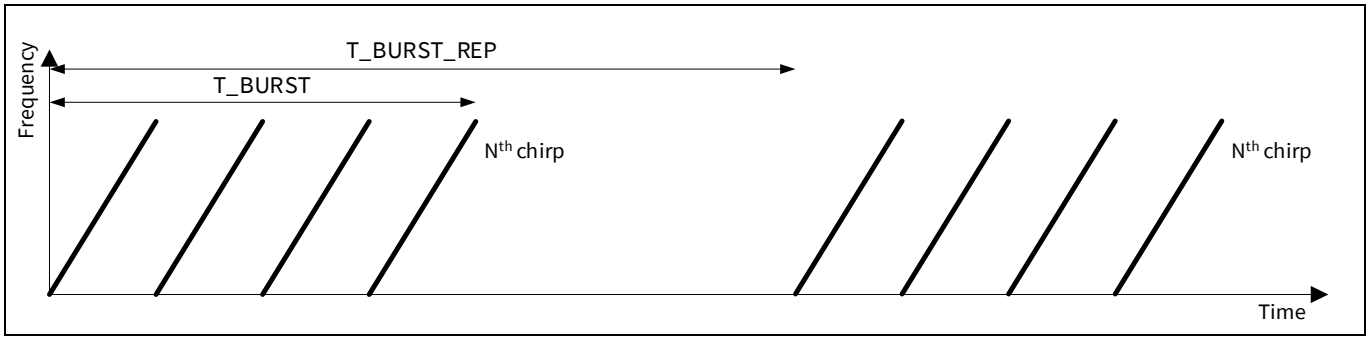


Figure 48 Burst

A radar frame is the complete set of data collected by the FMCW radar system during a single measurement cycle. It comprises multiple bursts, each containing several chirps, see Figure 49. The radar frame is the standard unit of data processing in FMCW radar, providing a snapshot of the environment that can be used to determine the target range, Doppler and angle information. The frame time represents the total time required to transmit a complete set of radar signals, including all bursts and chirps. Frame time together with frame repetition time defines the duty cycle of the radar which is critical for the overall power consumption of the radar system.

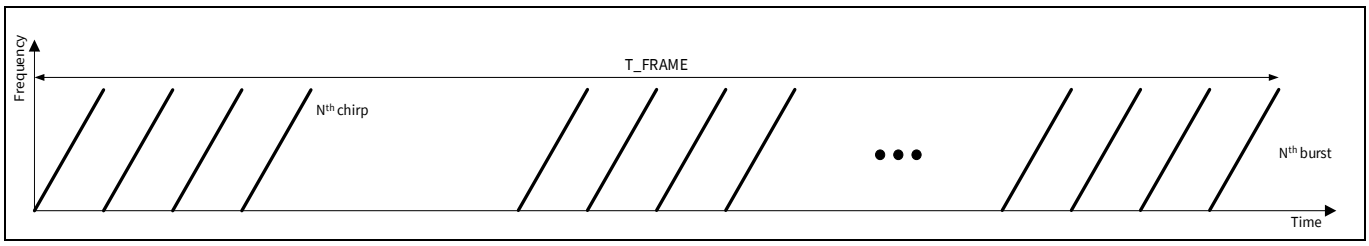


Figure 49 Frame

6.1.1 Sequencer (SEQ)

The XENSIV™ BGT60CUTR13AIP device offers a wide range of flexibility to define a frame due to its embedded frame sequencer. It is technically possible to define frames of any length and any number of chirps, each with pre-defined parameters. The frame definition itself is done via 16-bit commands stored in a dedicated memory called the SEQ_BUFFER. Up to four different frame definitions can be stored within the SEQ_BUFFER memory and executed at any time, triggered either via software (SPI write command), via I/O pin or via the embedded general timer module (GTMR) clocked by the embedded low-power RC oscillator (RCOSC).

This flexibility enables you to run up to four different frame definitions with up to 8 different chirp settings for different use cases, such as presence detection, gesture sensing, or vital sign monitoring without re-configuring the device on the fly.

To ensure the cycle-precise execution of the frame, each command decoded by the frame sequencer is executed by a finite state machine (FSM) clocked by the system clock (see Section “System clock”). The FSM ensures the deterministic behavior of the device, maintaining consistent timing of sequences regardless of the point in time or the trigger source.

The sequencer memory can hold up to 1 kbyte data (512 instructions) and is mapped to the address `0x3000 – 0x33FF`. The SEQ_BUFFER memory is connected to the on-chip bus so that it can be programmed by software via SPI.

6.1.1.1 Overview

Each sequence is defined by a start address configured via the register SEQ#_CTRL.START_ADDR_OFFSET. Multiple sequence trigger sources are available which can be individually enabled or disabled per sequence. The software trigger option (CLK_WUC.SEQ#_TRIG) cannot be disabled so that a software trigger from host is always possible. The following options for trigger sources are available:

- General timer (SEQ#_CTRL.GTMR#_EN)
- External trigger via DIO# (SEQ#_CTRL.TRIG#_EN)
- Motion detection event via ULP (SEQ#_CTRL.MD_EN)

Note: If multiple trigger sources are enabled and multiple triggers are received within the same system clock cycle (SYS_CLK), a hard-coded priority with SEQ0 having the highest priority and SEQ3 having the lowest priority is implemented.

The actual sequence status whether a sequence is active and running or not can be checked by software at any time by reading SEQ#_STAT.STATE. Furthermore, there is an option to generate an interrupt once a sequence execution is over (SEQ#_DONE, IR bit 8-11).

If required by the application, another sequence can immediately be started once a sequence is over. This feature can be enabled via SEQ#_CTRL.NEXT_SEQ_EN and SEQ#_CTRL.NEXT_SEQ_SEL.

Note: If the same sequence is selected, it results in infinite repetition, which can only be stopped by either clearing the bit NEXT_SEQ_EN or performing a reset.

The overall sequence counter can be read by software via SEQ#_STAT.SEQ_CNT and, if necessary, cleared via SEQ#_CTRL.SEQ_CNT_CLR. If the maximum counter value is reached, it starts again from zero.

6.1.1.2 Commands

There are four major types of commands, each defined by its own opcode and encoded by the two least significant bits (LSB) of the 16-bit command. Besides the structural commands, all commands are executed by the FSM and need at least one system clock (SYS_CLK) cycle for execution.

- 00_B: structural command
- 01_B: user configurable delay
- 10_B: configuration command
- 11_B: chirp command

Structure

This command type is used to structure the commands within the SEQ_BUFFER memory. As the memory needs to retain the data, it must be powered up all the time. To keep the overall power low, the size of the memory is limited to 1kB (512 commands). To be able to store up to four pre-configured sequences with only 512 commands, it is inevitable to re-use already defined sub-sequences and repeat certain commands multiple times. Therefore, the device uses structures like function calls, returns and loops.

Note: There is a maximum limit of nested loops of four to limit the size of the stack.

Delay

Sequences typically consist of two parts, one part to power-up the analog modules needed for the chirping and the second part consisting of the actual frame. Both require certain delays as the execution shall take place at different points in time. The smallest delay is one single-system clock cycle (SYS_CLK) and the largest goes up to milliseconds. This is realized by splitting the timer into an integer part called a multiplier (MUL) and an exponent (EXP) part similar to the floating-point number format.

Note: The additional system clock (SYS_CLK) cycle is always required to execute the command. Therefore, a delay of 1000 clock cycles results in a pause of 1001 clock cycles.

Configuration

Another command type is used to configure the analog components required for chirping, such as the ABB, ADC, and PLL. However, depending on the use case, not all available receive antennas and channels may be required. By enabling the required channels and their dedicated LDOs only, you can tailor the power consumption of their application, ensuring optimal efficiency. Each analog module comes with its own set of configuration commands, providing the flexibility to customize power usage based on specific application needs. This not only offers the advantage of application-specific power customization but also paves the way for future optimizations in the module's startup routines.

Note: For power optimization, the startup routines of RF+ABB, ADC, and PLL are executed in parallel and handled by a combined startup routine which takes around 135 µs.

Chirp

For frame definition, the most important command type is the chirp command. It defines parameters for the frequency ramps, the chirp modulation (such as binary phase-shift keying, BPSK), and the chirp type.

To provide maximum flexibility in chirp configuration, each chirp command can select:

- **One of eight freely configurable shape indices (SH_IDX):** Each shape index defines parameters, such as start frequency, slope, ramp time, and number of samples, configured via the `SH_IDX#_*` registers
- **One of eight freely configurable channel set indices (CS_IDX):** Each `CS_IDX` defines parameters, such as enabled transmit (TX) and receive (RX) antennas, anti-aliasing filter (AAF) settings, gain settings, and high-pass filter (HPF) configurations, set via the `CS_IDX#_*` registers
- **One of two freely configurable ADC indices (ADC_IDX):** Each `ADC_IDX` defines parameters, such as the decimation factor (oversampling) and the sample rate divider value, configured via the `ADC_IDX#_*` registers
- **One of two freely configurable sensor ADC indices (SADC_IDX):** Each `SADC_IDX` defines parameters, such as the decimation factor (oversampling), sample rate divider value, and sensor selection, configured via the `SADC_IDX#_*` registers

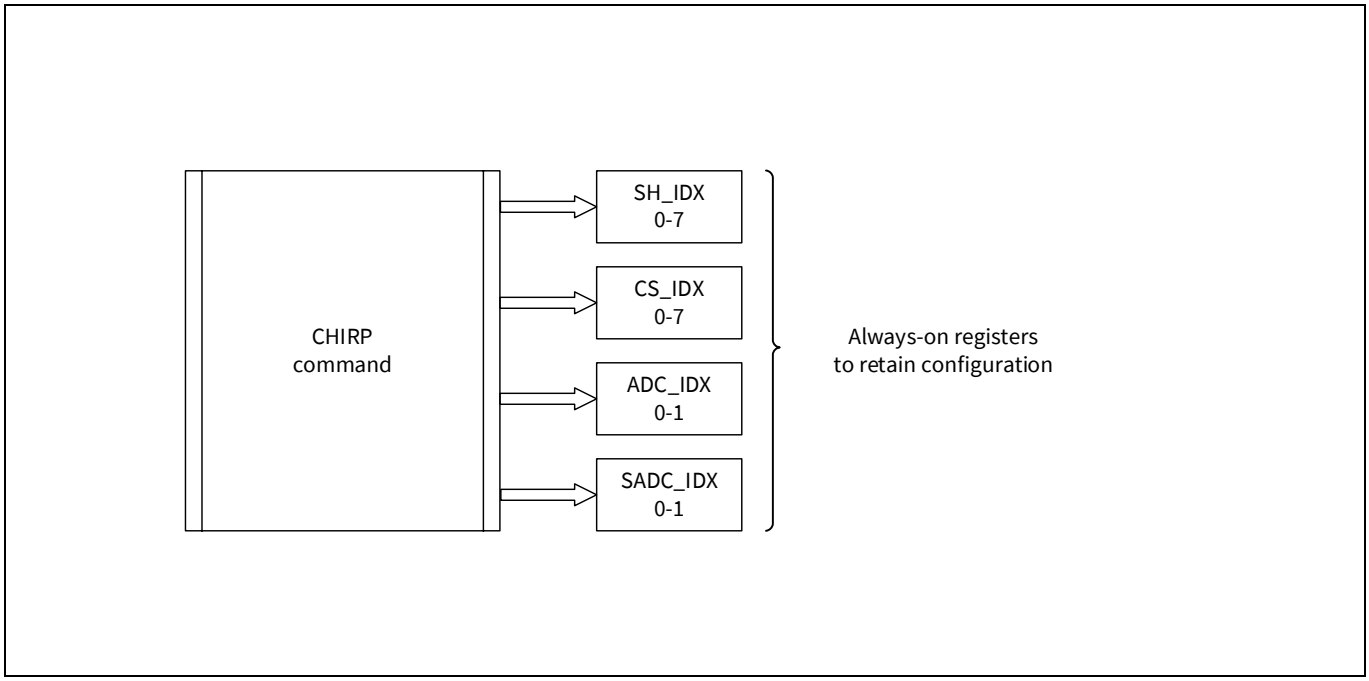


Figure 50 Chirp configuration

With these four types of commands depicted in Figure 50, it is possible to define a sequence containing the startup routine of each analog module, the user defined frame definition, and the power-down routine of the enabled analog modules.

6.1.1.3 Memory

All commands are stored in the sequence buffer memory (SEQ_BUFFER) which is accessible via SPI. As the overall system has a 32-bit data bus, two commands are stored together within one single 32-bit memory word. The SEQ_BUFFER is mapped to the address range `0x3000 – 0x33FF` of the on-chip bus. Figure 51 shows the command structure inside the memory.

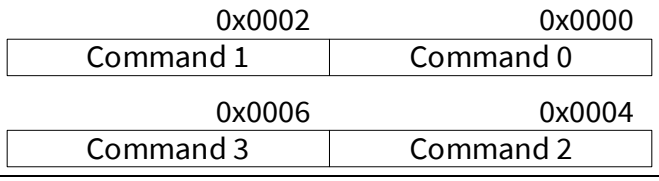


Figure 51 Sequence buffer memory word

Up to four different sequences can be stored within the SEQ_BUFFER, each identified by its unique offset address which is configurable via `SEQ#_CTRL.START_ADDR_OFFSET`. A typical sequence organization can be found in Figure 52. The sequence start address offset is byte aligned.

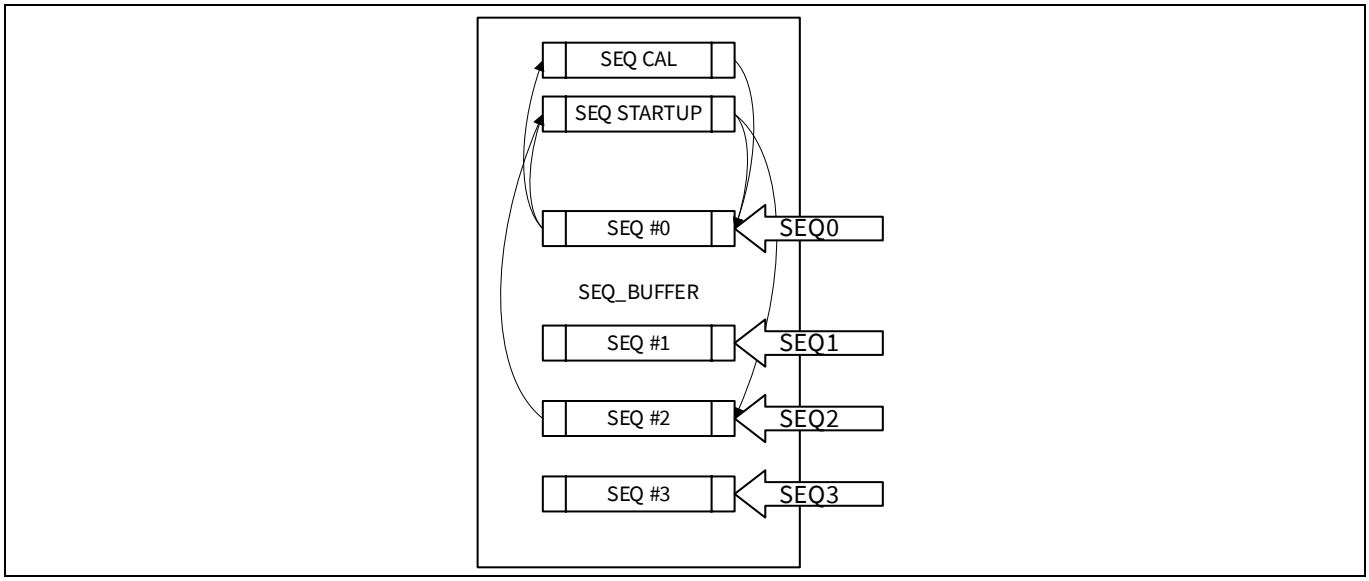


Figure 52 Sequence buffer memory – sequence organization

If multiple sequences are used, one sequence could be optimized for a presence detection, whereas a second sequence could be optimized for short range or far range frames. It is possible to jump to another address and execute commands at this position. Afterwards, when the return command of such a sub-routine (function) is reached, the original sequence is processed further. The same function can be used by multiple sequences to ensure efficient use of the memory size.

6.1.1.4 Error handling

FSM and sequencer inputs are provided via commands which are programmed by software. Using invalid commands, incorrect order of commands within sequences, or improper timing and scenarios may lead the sequencer to enter an invalid operational state. Those invalid conditions are caught by hardware and indicated by status bits (`SEQ#_STAT.*`). All error status is logically OR combined and flagged by a sequencer error status bit in the interrupt router to notify the host (bit 23). In some situations, the active sequence is stopped to prevent unpredictable behavior of the device.

The following errors are caught and reported by hardware:

- **JSE_ERR**

  The jump stack empty error is generated if there is a function return without any jump address on the jump stack. This could be the case if a return command is used without a jump command. The active sequence is stopped
- **JSF_ERR**

  The jump stack full error is generated if more than four nested jumps are used. The active sequence is stopped
- **LSF_ERR**

  The loop stack full error is generated if more than four nested loops are used. The active sequence is stopped
- **ST_ERR**

  The sequence trigger error is generated if there is a sequence running (active) and a further sequence trigger is received. As only one sequence can be executed at the same time this causes an error, and the new sequence trigger is ignored. The active sequence is not stopped

- **UCA_ERR**
  The unaligned command address error is generated if an address pointer (start address or jump address) uses an invalid command address, such as odd addresses like 0x1 or 0x3. In such cases, the LSB (bit 0) of the address is ignored (half word aligned addresses). The active sequence is not stopped
- **RO_ERR**
  The reserved opcode error is generated if a reserved opcode is used. Such commands are treated as a NOP (no operation) and are ignored. The active sequence is not stopped

6.1.1.5 Debugging and testing

For debugging and test purposes, there is the option to stop the sequencer command execution at each command like a breakpoint. The finite state machine (FSM) is kept on hold in the intended step (command) and if required, at the correct clock cycle (delay or chirp command).

This feature can be used to perform measurements at specific frequencies in the continuous wave (CW) mode. For this, the FSM stops at the chirp command where the PLL is set to a specific frequency and no frequency increase (continuous wave). Now multiple samples can be triggered via the `ADC_IDX#_CTRL.TRIG`.

To enter the debug mode of the sequencer, the debug mode must be enabled via `SEQ_TEST_DBG.EN`. Based on the number of commands in the sequence, the right step (command) number must be programmed to the `SEQ_TEST_DBG.STEP_NUM` register. There is also the option to select the required system clock cycle where the FSM will stop by programming the number of clock cycles to `SEQ_TEST_DBG.TMR`.

To verify if the device stopped at the right step, the command can be read back by the host via `SEQ_TEST_DBG_STAT.CMD`.

6.1.2 Chirp timing

All timings related to a chirp are highly configurable to adjust the slope of the PLL ramp, the effective bandwidth, and also the transmitted bandwidth. All programmable timers start with the chirp command and are running in parallel. Once the period of a timer expires, the corresponding signal is set or the respective action such as a sampling start is initiated. Once the timer for the main chirp time (T_CHIRP) expires, the next command is executed by the finite state machine (FSM). The most important timings are:

- **T_START** (`SH_IDX#_TR_START.MUL`)
  The PLL start timer can be used to delay the ramp processing by an additional time to let analog signals settle properly in the RF domain
- **T_RAMP** (`SH_IDX#_TR_RAMP.MUL`)
  The PLL ramp timer defines the overall duration of the ramp time
- **T_FSTDN** (`SH_IDXx.TR_FSTDN`)
  The PLL fast down timer defines the time the PLL uses for a fast flyback to the start frequency
- **T_END** (`SH_IDX#_TR_END.MUL`)
  The PLL end timer can be used to delay the PLL ramp processing after the end frequency is reached
- **T_TXEN** (`SH_IDX#_TR_TXEN`)
  The transmit enable timer can be used to enable the power amplifier once the PLL reaches its linear region of the ramp
- **T_ABBRST** (`SH_IDX#_TR_ABBRST`)
  The ABB reset timer can be used to keep the ABB in reset if the PLL is not in the linear region and no ADC sampling is ongoing

- **T_SAMPLE** (SH_IDX#TR_SAMPLE)
  The sampling timer is used to delay the ADC sampling start until the PLL is in a linear region

Note: Only the timings T_START, T_RAMP, T_FSTDN, and T_END have an impact on the chirp timing.

The total chirp timing can be calculated using Equation 5.

$$T_{Chirp} = TR_{START} + TR_{RAMP} + (TR_{FSTDN} + 1) + TR_{END} + 16$$

Equation 5 Chirp timing

Figure 53 provides a visual representation of the relationships among all the previously mentioned timers.

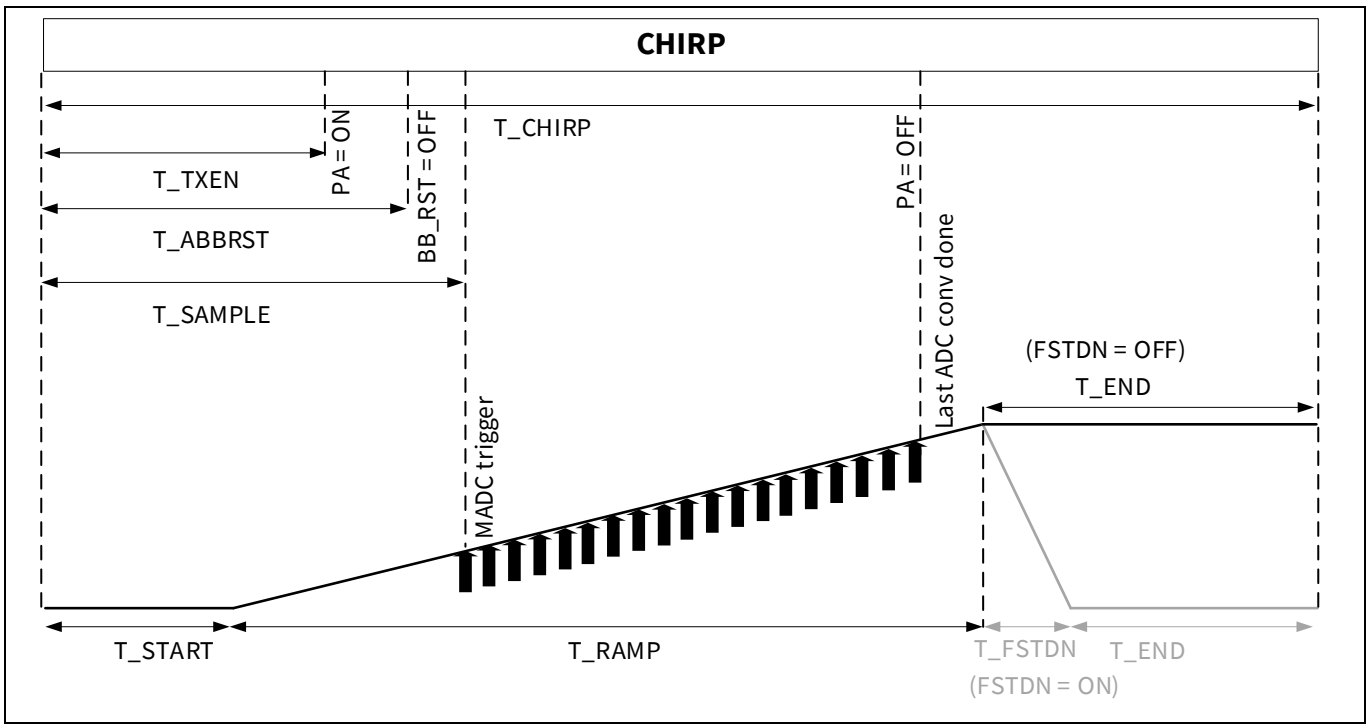


Figure 53 Chirp timing

6.1.3 Extended bandwidth (interlaced) chirps

There is an option to run multiple chirps sequentially with an increased offset from chirp to chirp (stepped chirp) as shown in Figure 54 to increase the total bandwidth. For this, the interlaced chirp (IC) bit of the chirp command must be set and the additional frequency offset from chirp to chirp must be defined via the register SH_IDX#_OFFSET.FREQ () in the format Q2.20 with 1 sign bit, 2 integer bits, and 20 fractional bits.

$$SH_{IDX}\#_{OFFSET}.FREQ = 2^{20} \times \frac{\Delta f_{PLL}}{4 \times f_{PLL_{REF_{CLK}}}}$$

Equation 6 PLL (chirp) offset frequency

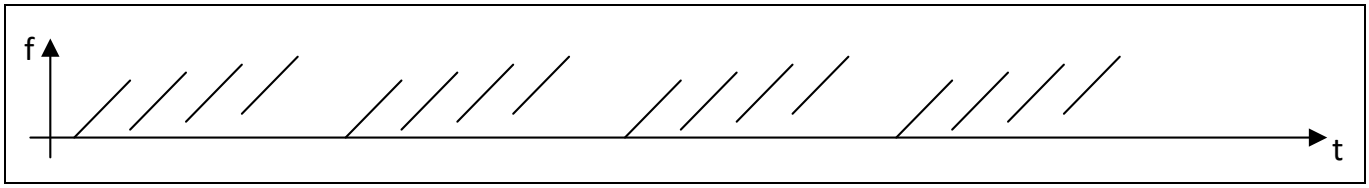


Figure 54 Extended bandwidth (interlaced) chirps

Note: There is also an option to apply a negative offset frequency and use down-chirps instead of up-chirps.

6.2 Data flow

The signal received by the RX antenna is down converted, shaped by the analog baseband (ABB), and converted into a digital representation of the same. Later, it is optionally decimated and averaged with signals of previous chirps before it is finally stored in the FIFO. For data handling and structuring, there is an option to enable a data header which provides additional meta data and sensor data to the application software. Figure 55 shows the data flow block diagram of XENSIV™ BGT60CUTR13AIP.

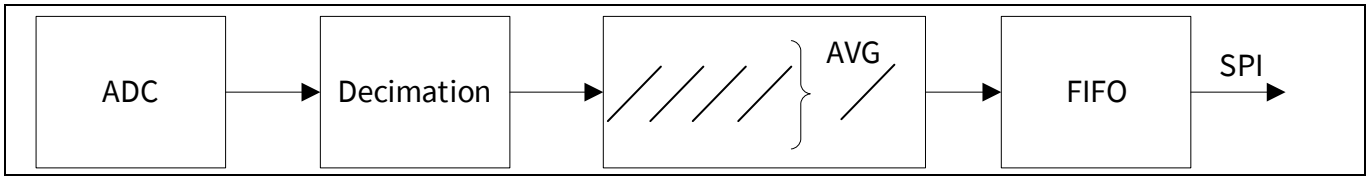


Figure 55 Block diagram of BGT60CUTR13AIP data flow

6.2.1 Oversampling and decimation

The resolution of the ADC is 12 bits + 4 fractional bits (see ADC calibration). However, due to noise and inaccuracy losses, the effective number of bits (ENOB) is 10.25 = ((63.5 dB - 1.76 dB)/6.02 dB).

This can be improved by using a higher oversampling factor and applying averaging and decimation to the ADC digital output. Each averaging and decimation by a factor of four results in a better signal-to-noise ratio (SNR) of +6 dB which is effectively an improvement by one bit ((63.5 dB + 6 dB -1.76 dB) / 6.02 dB = 11.25).

The averaging is applied on groups of n consecutive samples, where n = $2^{ADC_IDX\#_CTRL.DF}$. Figure 56 shows an example of how the averaging is applied with $ADC_IDX\#_CTRL.DF = 1_D$.

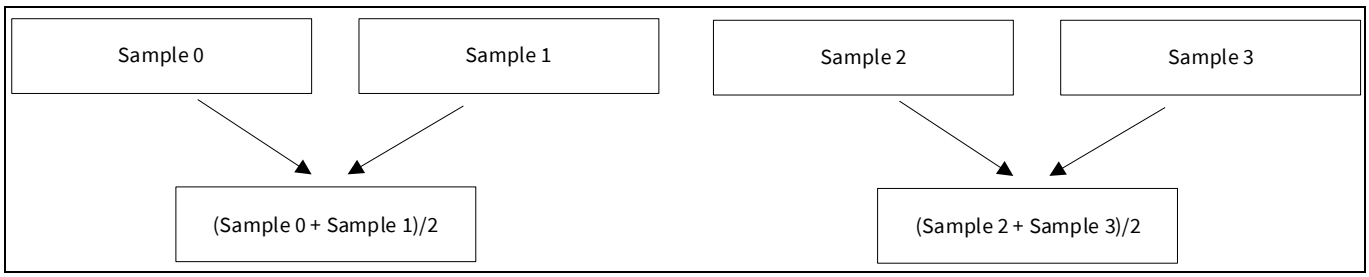


Figure 56 Averaging and decimation example with two samples

In Figure 57, the intermediate frequency signal is oversampled with an oversampling ratio of four resulting in a +6 dB better SNR. The individual ADC conversion results are affected by noise. However, the averaged conversion is much closer to the real IF signal.

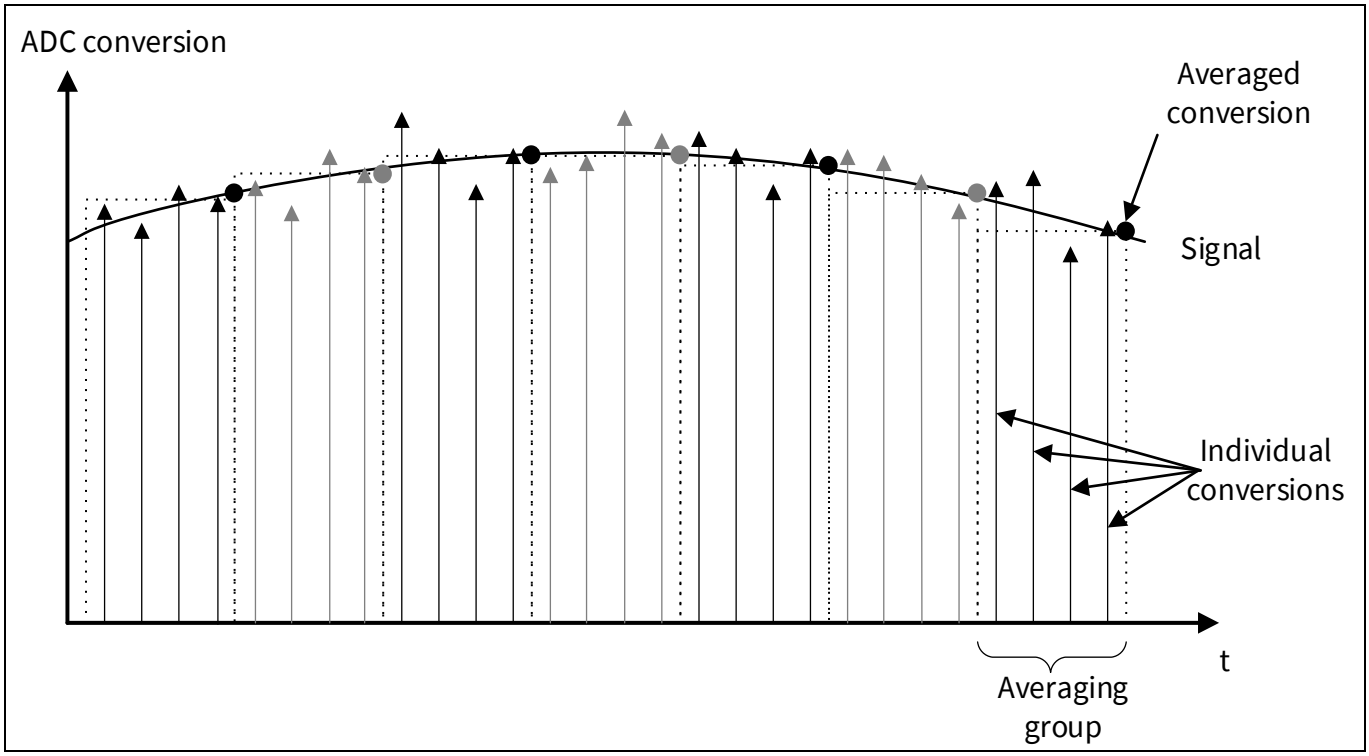


Figure 57 Intermediate frequency signal with oversampling

Note: The total number of ADC conversions is defined by `SH_IDX#_SAMPLE.NUM` x 2^`ADC_IDX#_CTRL.DF`. However, due to the embedded decimation, only the number of samples defined by SH_IDX#_SAMPLE.NUM is stored in the FIFO or used for further processing.

6.2.2 Chirp averaging (AVG)

For some frames, it could be necessary to combine multiple consecutive chirps with one single chirp. To not stream out all ADC raw data and to perform such chirp averaging on the host controller, the device contains an embedded chirp averaging module. It enables you to average per RX channel up to 256 consecutive chirps with up to 1024 samples each, to one single chirp in time domain. Such a back-to-back chirp structure is called burst (shown in Figure 58) as its size can be defined for each sequence individually via the register `AVG_CTRL_SEQ#.NUM`. Each sample N of the chirp is accumulated with sample N of other chirps within the burst and divided by the burst size afterwards.

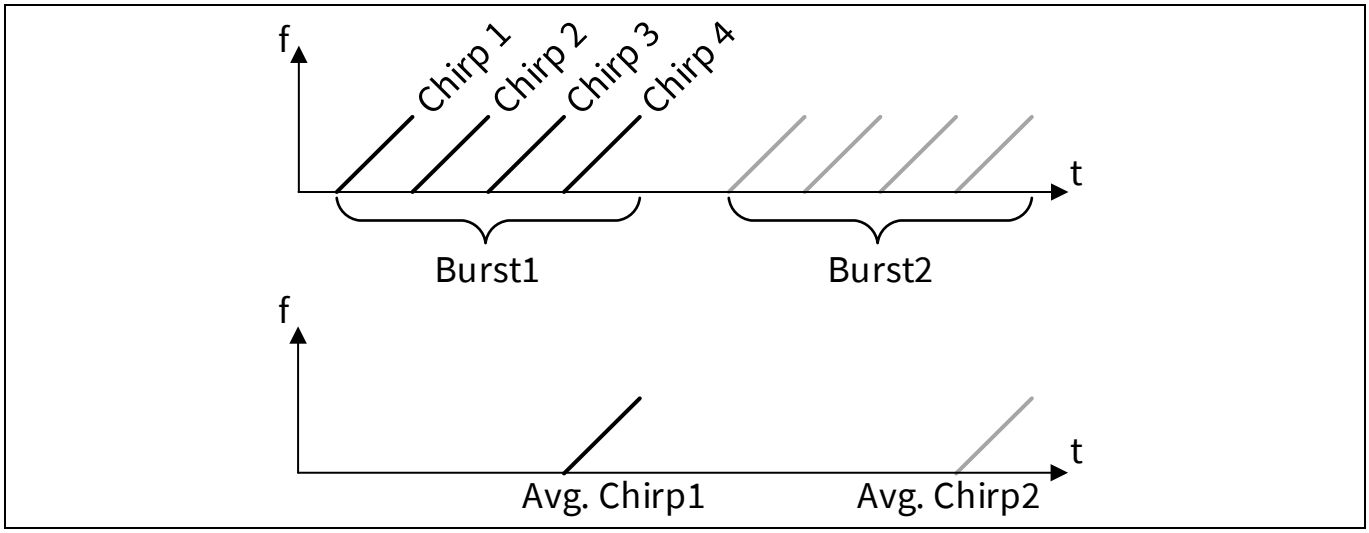


Figure 58 Chirp averaging

Each RX channel has its own dedicated local processing memory (RX_BUFFER) which is only powered-up during processing and powered-down afterwards to reduce the overall power consumption. Its power mode can be controlled with `AVG_CTRL.PM`.

6.2.3 First-in first-out (FIFO) – Data memory

All ADC data is organized in the FIFO with a 32-bit data width. A configurable fill level threshold (`FIFO_CTRL.THRESHOLD`) is used to generate an interrupt (IR bit 7) once the FIFO fill level exceeds this limit. Later, the data of the FIFO is read by software via SPI. The FIFO can store up to 49152 16-bit ADC samples and its read pointer is mapped to the address `0x4000` to `0x7FFF`.

Note: If a SPI burst read is used to read the FIFO, it must be ensured that the maximum number of 32-bit words of the burst is not exceeding 4096 words when started at address `0x4000`. Otherwise, the maximum address of the FIFO read pointer of `0x7FFF` is exceeded resulting in wrong data.

There is also an option to read the FIFO fill level at any time by software via the register `FIFO_STAT.LEVEL`. In addition to this, multiple status bits reflect the current status of the FIFO and indicate if the FIFO is full (`FIFO_STAT.FULL`), empty (`FIFO_STAT.EMPTY`) or if the FIFO fill level threshold is exceeded (`FIFO_STAT.TH`), as shown in Figure 59.

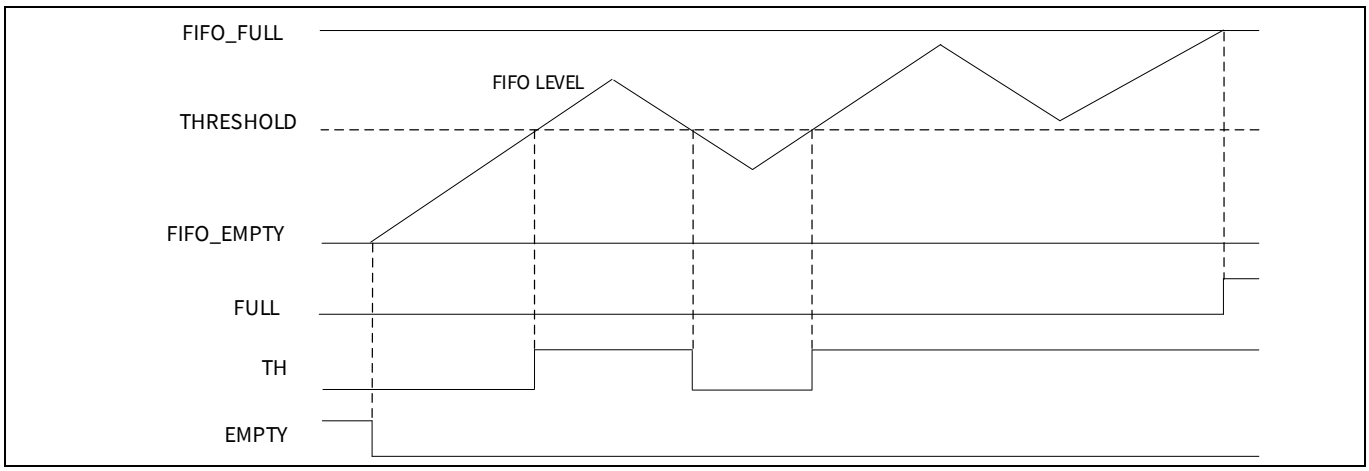


Figure 59 FIFO status

Underflow:
In case a FIFO underflow occurs, no SRAM read is performed and the software reads a 32-bit word with all ones. With this pattern, the host can still use the receiving data and only withdraw the useless data words with all ones (`0xFFFF`). The FIFO indicates an underflow by setting the corresponding bit in the interrupt router (bit 16).

Overflow:
In case a FIFO overflow occurs, the ADC data is not stored anymore and is lost. The FIFO indicates an overflow by setting the corresponding bit in the interrupt router (bit 17). Furthermore, the active sequence is stopped.

Threshold:
The bit TH is set while the FIFO fill level is above ('>') the programmed threshold value `FIFO_CTRL.THRESHOLD`. It is cleared when the fill level is equal ('=') or below ('<') the threshold value.

$(N_{SAMPLE} \times N_{RX})/2 - 1$

Note: In case a FIFO underflow or overflow occurs, error bits are set in the interrupt router (`FIFO_UF_ERR` (bit 16), `FIFO_OF_ERR` (bit 17)).

6.2.3.1 Data structure

Based on the number of active receiver (RX) channels and their respective ADCs, different data structures are applied inside the FIFO. As each application and use-case has an individual channel requirement, different combinations are possible.

- **One ADC:** ADC1 or ADC2 or ADC3
- **Two ADCs:** (ADC1 and ADC2) or (ADC1 and ADC3) or (ADC2 and ADC3)
- **Three ADCs:** ADC1 and ADC2 and ADC3

If multiple ADCs are active, then the samples are stored in ascending order; lowest ADC number sample is stored first, the next higher ADC number sample is stored next, and the highest ADC number is stored last. With that, the order of the samples is always defined and deterministic. Figure 60, Figure 61, and Figure 62 shows different number of active ADCs and their data organization inside the FIFO.

One ADC:

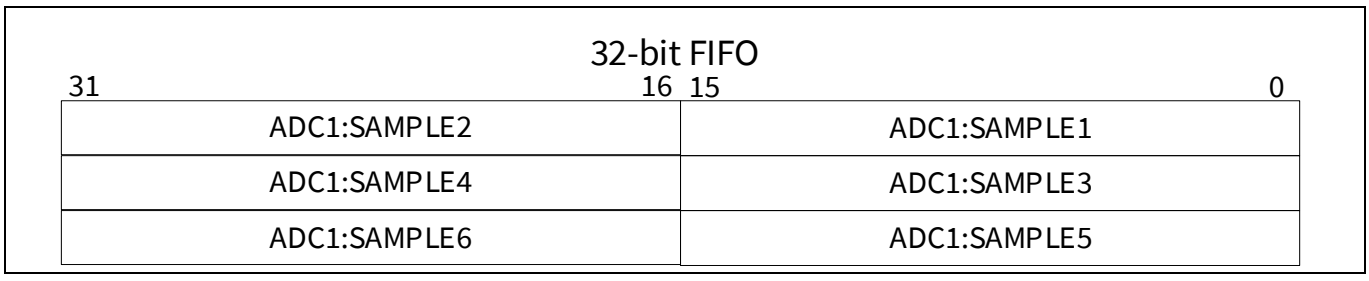


Figure 60 Data structure of one ADC

Two ADCs:

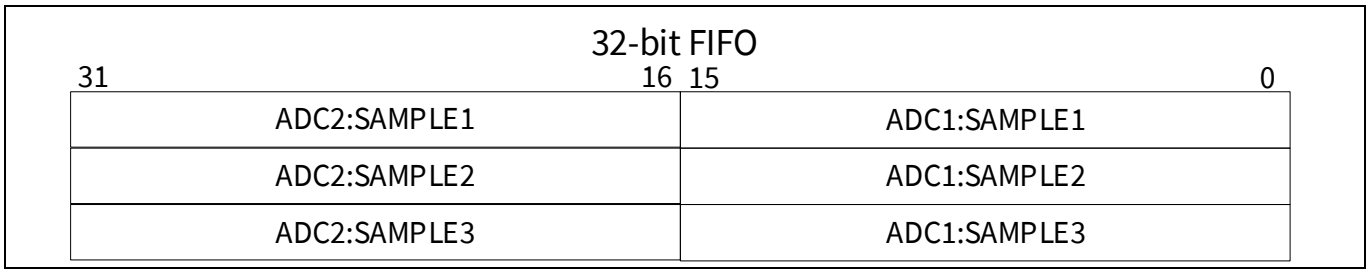


Figure 61 Data structure of two ADCs

Three ADCs:

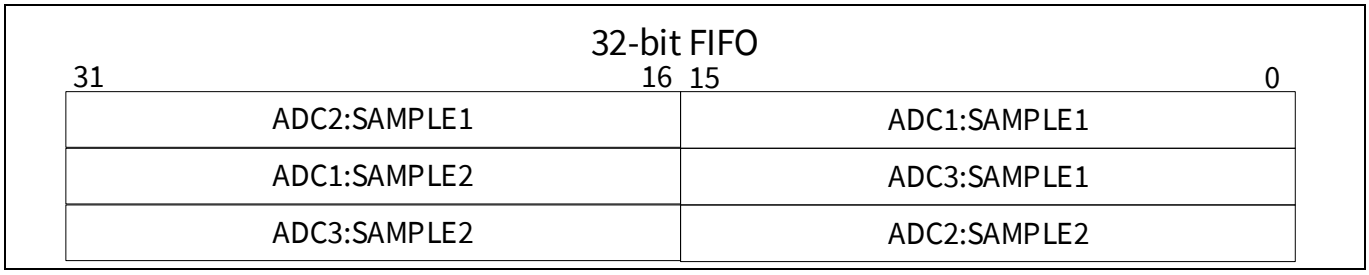


Figure 62 Data structure of three ADCs

6.2.3.2 Data header

For easier data handling, the device provides the feature to insert a header with additional meta and sensor data in front of chirp or sequence data inside the first-in first-out (FIFO) memory. There are two options available to choose at which position the additional header information is inserted. The header location can be selected via `FIFO_CTRL.HEAD_LOC`.

- **Sequence** (0_B: SEQ)

  The header data is inserted before the first ADC data at the beginning of each sequence
- **Chirp** (1_B: CHIRP)

  The header data is inserted before each chirp data in the FIFO and therefore, available multiple times per sequence. If the chirp averaging feature is enabled (`AVG_CTRL_SEQ#.NUM > 1`), then only one header per burst is inserted

Note: The optional chirp counter will increase in steps of the configured burst size of the sequence.

The header itself can be customized according to the requirements of the application software. The following options are available and can be selected via `FIFO_CTRL.HEAD*`.

- **Synchronization** (`HEAD_SYNC_EN`)

  A 32-bit sync word with all zeros added (`0x00000000`)
- **Sensor ADC results** (`HEAD_SADC_EN`)

  The value of `SADC_IDX0_STAT.RESULT` and `SADC_IDX1_STAT.RESULT` are added
- **Counter values** (`HEAD_CNT_EN`)

  The respective sequence number (SEQ#), the sequence counter (SEQ_CNT), and the global chirp counter (CHIRP_CNT) values are added. All counters start from zero for the first chirp or sequence

Data header format is shown in Figure 63.

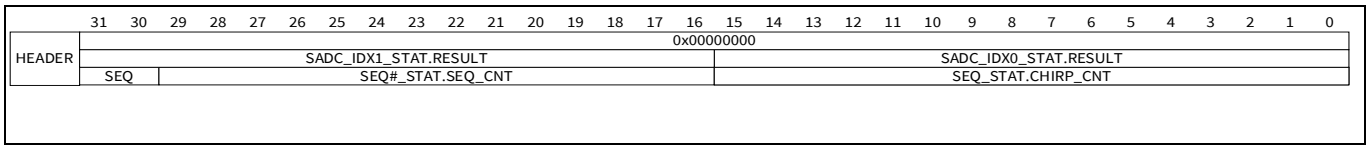


Figure 63 Data header format

Note: The data storage described in this chapter only applies to unprocessed data. If data forwarding of processed data is enabled, the data structure is different, see Section Data forwarding (FIFO).

7 Presence detection (PD)

The presence detection functionality provides the capability to detect presence in the field of view of the device. The intention of this feature is to reduce the overall power consumption of the system and only wake up the host via an interrupt if a target is detected in the field of view (FoV) of radar. To handle different use cases, two on-chip signal processing chains namely ultra-low power (ULP) and range Doppler (RD) modes are implemented in the device.

7.1.1 Overview

Optimizing power consumption and presence detection accuracy:

To achieve the perfect balance between power consumption and false alarm rate, Infineon has developed two on-chip signal processing chains for the presence detection (PD). These innovative solutions enable the device to operate at the lowest power while still delivering accurate presence detection capabilities.

ULP Sequence (SEQ1): Ultra-low power (ULP) frames:

This sequence features frames containing only two bursts separated in time by a small delay, which ensures extremely low power consumption. This on-chip processing chain is highly effective in identifying potential motion and quasi static targets in the FoV of radar.

RD Sequence (SEQ2): Range Doppler (RD) frames:

This sequence features frames that typically contain a minimum of 8 bursts, which enables the generation of precise range Doppler images (RDI). While these frames consume higher power compared to ULP sequence, they provide detailed range and Doppler information, allowing for accurate motion and presence detection. To minimize the power consumption, this on-chip processing chain can be configured to run upon detecting a motion by the ULP sequence.

7.1.2 Operation modes

This device offers flexible configuration options, allowing it to operate in standalone ultra-low power (ULP) mode or range Doppler (RD) mode, as well as combining these modes to meet various use cases. For instance, it can be set to use RD mode for motion detection and ULP mode for detecting quasi-static targets. Additionally, the device can optimize power consumption and motion detection accuracy by combining ULP and RD sequences, making it suitable for a broad range of applications.

7.1.2.1 Ultra-low power (ULP)

The ultra-low power (ULP) mode enables motion and quasi-static target detection with only two Bursts. ULP is meant to be used to achieve minimal standby power consumption by still being selective in range and optionally angle. ULP can be combined with a more accurate Range Doppler (RD) processing in case a motion has been detected. Figure 64 shows the ULP processing pipeline with optional and mandatory algorithm selection.

Since acquisition and data processing are separated on the chip, it is important to configure the frame according to the requirement for the processing pipeline. However, there are still multiple configuration options possible. The ULP processing is executed when two bursts are generated by the sequence and therefore, `MD_ULP_CTRL.NUM_BURST` must be configured to 2_D. The number of samples per chirp must be a power of 2 with a minimum of 64_D and a maximum of 512_D which is defined in the shape index of the respective sequence (`SH_IDX#_SAMPLE.NUM`) and configuration register for ULP (`MD_ULP_CTRL.RFFT_SIZE`).

To achieve a better signal-to-noise ratio (SNR), the embedded chirp averaging (AVG) function can be used. In this case, the frame must generate the correct number of chirps per burst, and the chirp averager function must be configured via `AVG_CTRL_SEQ#.NUM` accordingly.

ULP processing can be enabled via `MD_ULP_CTRL.EN`. The respective sequence can be selected via `MD_ULP_CTRL.SEQ_SEL`. Active channels for processing can be enabled via `MD_ULP_CTRL.RX#_ACT`. Typically, in ULP mode, receiver channel one (RX1) is processed for motion and presence detection.

The overall ULP processing pipeline is presented in Figure 64. Each processing block is described in more detail in Section Algorithm selection.

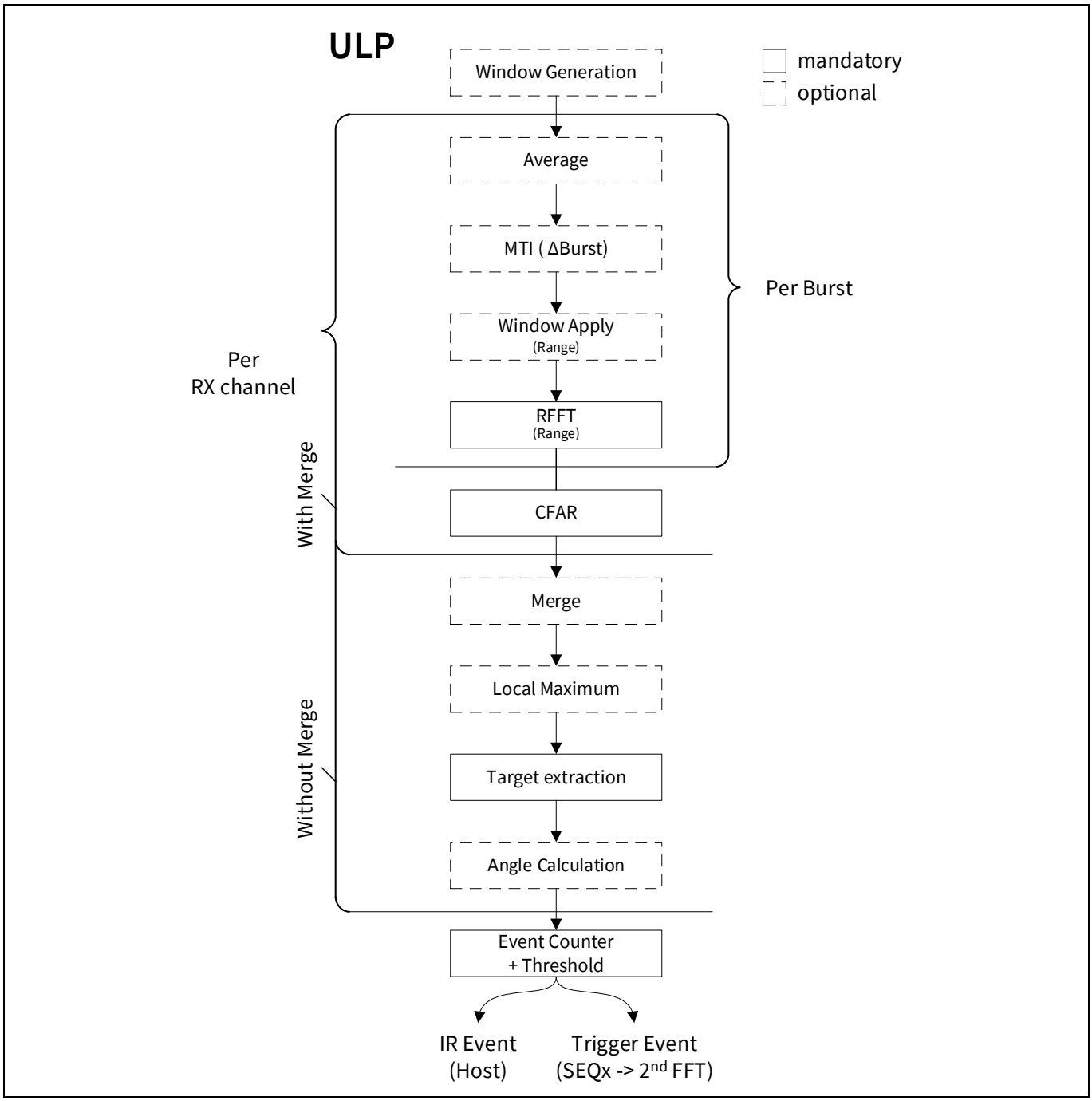


Figure 64 ULP processing pipeline

As each ULP frame requires two bursts separated in time, the burst repetition time (T_BURST_REP) can be modified to adjust the sensitivity of motion.

For power efficiency there are two options to define the ULP frame:

- **Option 1:** For burst repetition times (T_BURST_REP) with less than 1 ms, the most power-efficient implementation is to define the complete ULP frame within one sequence with the consequence of keeping the complete system running between both bursts
- **Option 2:** For burst repetition times (T_BURST_REP) with more than 1 ms, the most power-efficient implementation is to split the ULP frame into two sequences containing only one single burst and triggering it twice using GTMR module. This way the system can enter the most power-efficient power state DeepSleep in between. For that option, the data is retained in the RX_BUFFER SRAM until the ULP processing is over

Option 1 of an ultra-low power frame is shown in the Figure 65 and sequence timing in Figure 66.

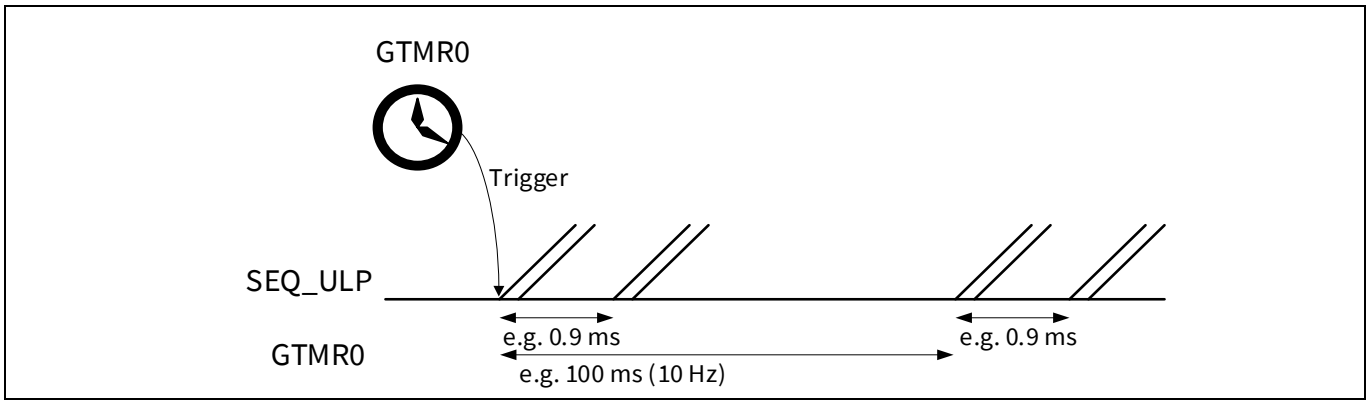


Figure 65 Option 1: Ultra-low-power frame with one sequence

Note: When ULP presence detection is enabled, ensure that the RCOSC is active as the GTMR# requires the RCOSC_CLK.

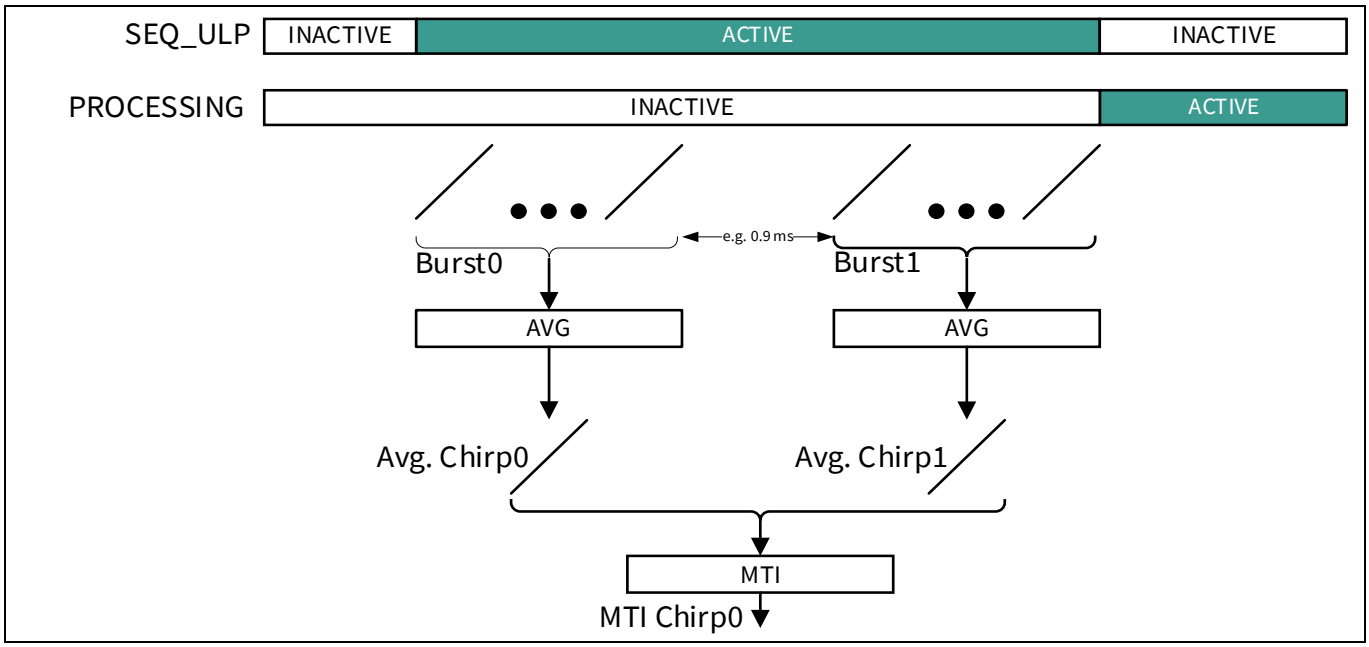


Figure 66 Option 1: Ultra-low-power sequence timing

Code listing 7 ULP register configuration

```
MD_ULP_CTRL.EN=1B
MD_ULP_CTRL.SEQ_SEL=1D // ULP processing starts after SEQ1
MD_ULP_CTRL.NUM_BURST=2D
MD_ULP_CTRL.RX1_ACT=1B // channel RX1 used for processing
GTMR0_CTRL.EN=1B
GTMR0_CTRL.REP=0D // endless repetitions
GTMR0_CTRL.SRC_SEL=15D // timer start by software trigger
GTMR0_TIMER.VALUE=8000D // 100 ms at 80 kHz RCOSC
SEQ1_CTRL.GTMR0_EN=1B // GTMR0 starts sequence 1
```

Note: The initial GTMR0 trigger must be done by software via `GTMR0_CTRL.TRIG=1B`.

Option 2 of an ultra-low power frame is shown in Figure 67 and the sequence timing in Figure 68.

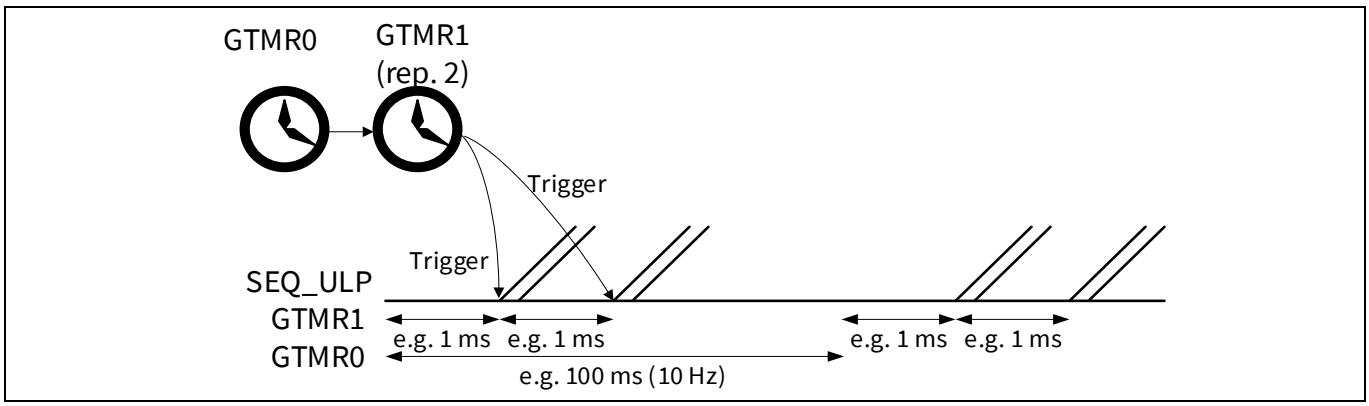


Figure 67 Option 2: Ultra-low-power frame split into two sequences

Note: When ULP presence detection is enabled, ensure that the RCOSC is active as the GTMR# requires the RCOSC_CLK.

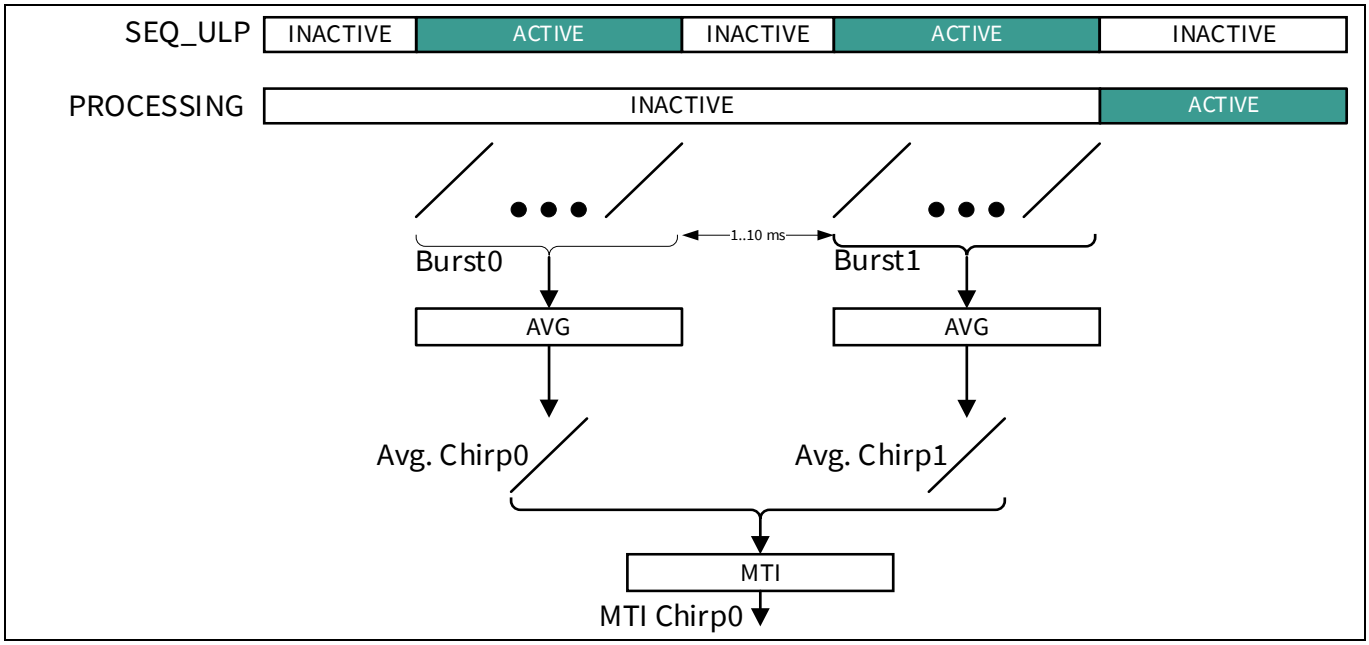


Figure 68 Option 2: Ultra-low-power sequence timing

Code listing 8 ULP register configuration

```
MD_ULP_CTRL.EN=1B
MD_ULP_CTRL.SEQ_SEL=1D // ULP processing starts after SEQ1
MD_ULP_CTRL.NUM_BURST=2D
MD_ULP_CTRL.RX1_ACT=1B // channel RX1 used for processing
GTMR0_CTRL.EN=1B
GTMR0_CTRL.REP=0D // endless repetitions
GTMR0_CTRL.SRC_SEL=15D // timer start by software trigger
GTMR0_TIMER.VALUE=8000D // 100 ms at 80 kHz RCOSC
GTMR1_CTRL.EN=1B
GTMR1_CTRL.REP=2D // two repetitions
GTMR1_CTRL.SRC_SEL=4D // timer start by GTMR0
GTMR1_TIMER.VALUE=80D // 1 ms at 80 kHz RCOSC
SEQ1_CTRL.GTMR1_EN=1B // GTMR1 starts sequence 1
```

Note: The initial GTMR0 trigger must be done by software via `GTMR0_CTRL.TRIG=1B`.

7.1.2.2 Range Doppler (RD)

Range Doppler (RD) processing can be used to detect motion with a minimum of 8 bursts. RD processing can be used to detect motion selective in range, velocity, and angle of arrival (AoA). In contrast to ULP, a range-Doppler image (RDI) is generated for a more accurate detection by calculating a second complex FFT (CFFT).

The number of bursts processed by the CFFT is configured by `MD_RD_CTRL.CFFT_SIZE` and must be configured to a power of 2 with a minimum of 8 bursts and a maximum of 64 bursts. The number of bursts acquired by an RD Frame is configured by `MD_RD_CTRL.NUM_BURST`.

Note: If MTI is enabled, the number of acquired bursts must be the value of `MD_RD_CTRL.CFFT_SIZE` incremented by 1. However, it is recommended to keep the MTI disabled during RD.

The number of samples per chirp shall be a power of 2 with a minimum of 16_D and a maximum of 512_D and is defined in the shape index of the respective sequence (`SH_IDX#_SAMPLE.NUM`) and configuration register for RD (`MD_RD_CTRL.RFFT_SIZE`).

The physical size of the RX_BUFFER memory is limited by 1024 words, six bytes each (three bytes for real and three bytes for imaginary data). Therefore, number of range bins multiplied by number of Doppler bins shall not exceed 1024 complex points.

To achieve a better signal-to-noise ratio (SNR), the embedded chirp averaging (AVG) function can be used. In this case, the frame must generate the correct number of chirps per burst, and the chirp average module must be configured via `AVG_CTRL_SEQ#.NUM`.

RD processing can be enabled via `MD_RD_CTRL.EN`. The respective sequence can be selected via `MD_RD_CTRL.SEQ_SEL`. Active channels for processing can be enabled via `MD_RD_CTRL.RX#_ACT`. Typically, in RD mode all three receiver channels are processed for motion and presence detection.

The overall RD processing pipeline is presented in Figure 69 with timing in Figure 70. Each processing block is described in more detail in Section Algorithm selection.

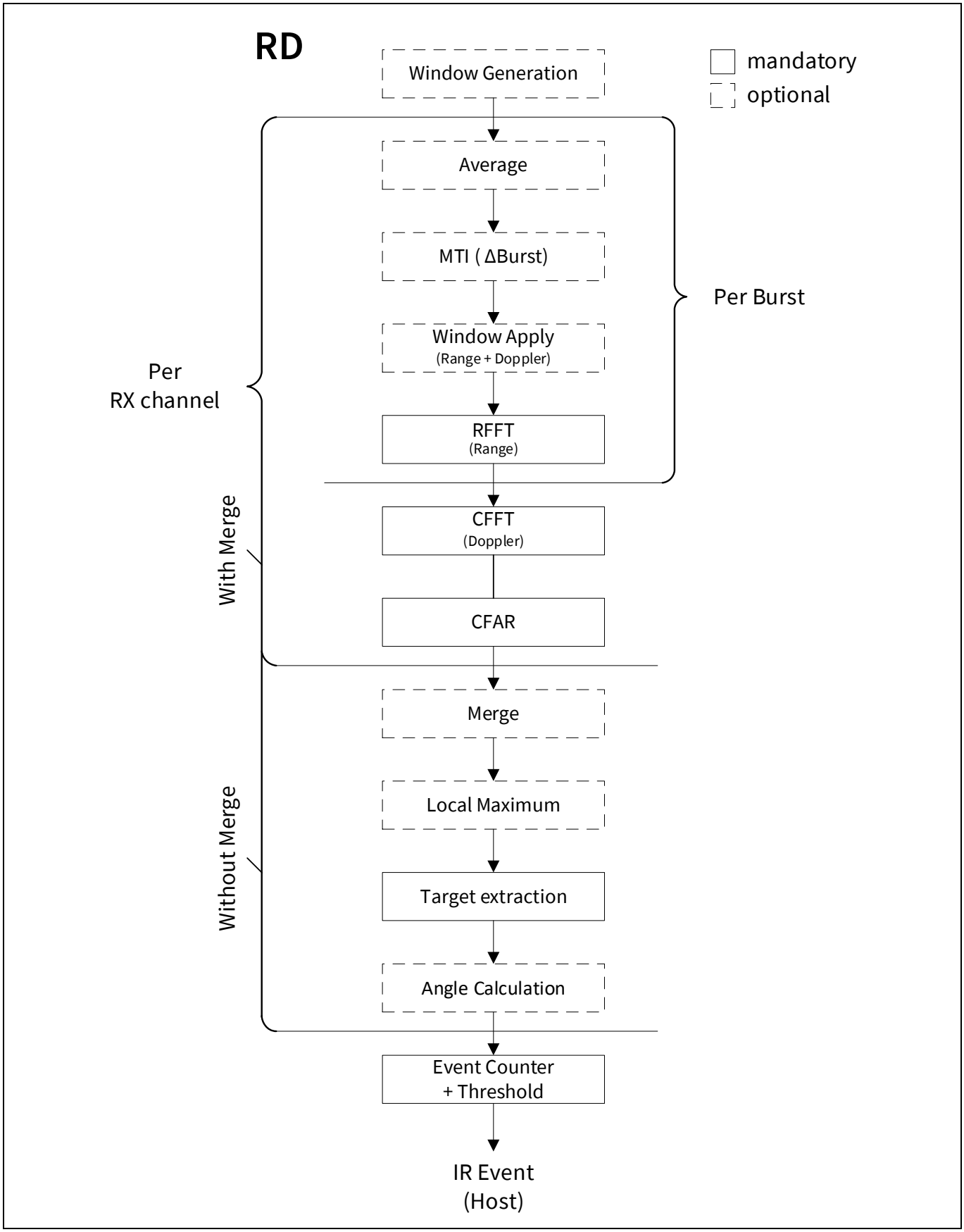


Figure 69 RD processing pipeline

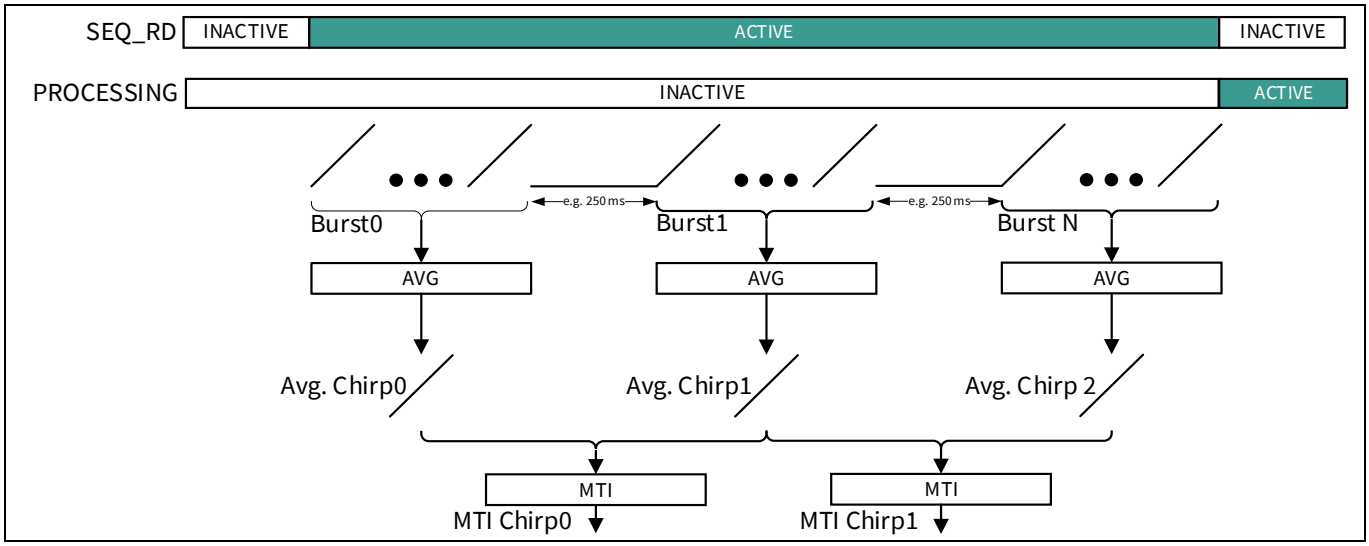


Figure 70 RD timing

Code listing 9 RD register configuration

```
MD_RD_CTRL.EN=1B
MD_RD_CTRL.SEQ_SEL=2D // RD processing starts after SEQ2
MD_RD_CTRL.RX1_ACT=1B // channel RX1 used for processing
MD_RD_CTRL.RX2_ACT=1B // channel RX2 used for processing
MD_RD_CTRL.RX3_ACT=1B // channel RX3 used for processing
MD_RD_ALG_CTRL.ANGLE_EN=1B // angle estimation enabled
MD_RD_FFT_CTRL.NUM_BURST=17D // after MTI 16 chirps
SEQ1_CTRL.MD_EN=1B //ULP motion detection event triggers SEQ_RD
```

7.1.3 Algorithm selection

ULP and RD processing consist of algorithm chains, see Figure 64 and Figure 69. Available algorithms are explained in detail in the following sections. In general, it is possible to enable and disable most algorithms at will, depending on the requirements. Keep in mind that not all possible combinations are useful. Limitations to algorithm selection are:

- ULP is limited to 1-dimensional frames

Some algorithms require other algorithms to be executed before to ensure the correct data format. These requirements are:

- Windowing: window coefficients must be generated beforehand, either by enabling Window Generation or by writing the window coefficients through SPI to the MD_BUFFER SRAM
- CFFT: requires RFFT
- CFAR: requires CFFT in RD sequences and RFFT in ULP sequences
- RX merge: requires CFAR
- Local maximum: requires CFAR
- Target extraction: requires CFAR
- Angle estimation: it requires target extraction and at least two active receiver channels
- Event counter: requires target extraction

Algorithms can be enabled or disabled with `MD_ULP_ALG_CTRL` or `MD_RD_ALG_CTRL`, respectively.

Note: It is advised to have RX merge and Local Maximum always enabled. Disabling it can cause targets to get reported multiple times.

7.1.3.1 Moving target indicator (MTI)

The moving target indicator (MTI) is used to remove static target information by calculating the delta between two subsequent averaged chirps in the time domain of the received signal. It is usually used for ULP based motion detection processing but can also be used for RD based motion detection. MTI operation reduces the number of averaged chirps during processing by one. After MTI the number of averaged chirps must be 1 for ULP and 8, 16, 32, 64 or 128 for RD sequences. This results in the requirement for the number of averaged chirps before MTI to be in the form $2^x + 1$ if MTI is enabled, else 2^x. MTI is enabled with `MD_ULP/RD_ALG_CTRL.MTI_EN`.

When the MTI DELTA is enabled (`MD_ULP_MTI_CTRL0.MODE=1B`, `MD_RD_MTI_CTRL0.MODE=1B`), it calculates the delta of two subsequent averaged chirps in the time domain by subtracting sample N of averaged chirp M from sample N of averaged chirp M-1, see Figure 71. As the first averaged chirp is stored temporarily within the RX_BUFFER memory, it is possible to split the two consecutive average chirps over two separate sequences. This allows to power-down the complete device after the first averaged chirp while the chip is in the DeepSleep state and the RX_BUFFER memory is kept active to retain the data. Once the second averaged chirp is received (such as ULP sequence with one averaged chirp triggered again), the MTI performs the delta calculation and triggers the next step in the processing pipeline.

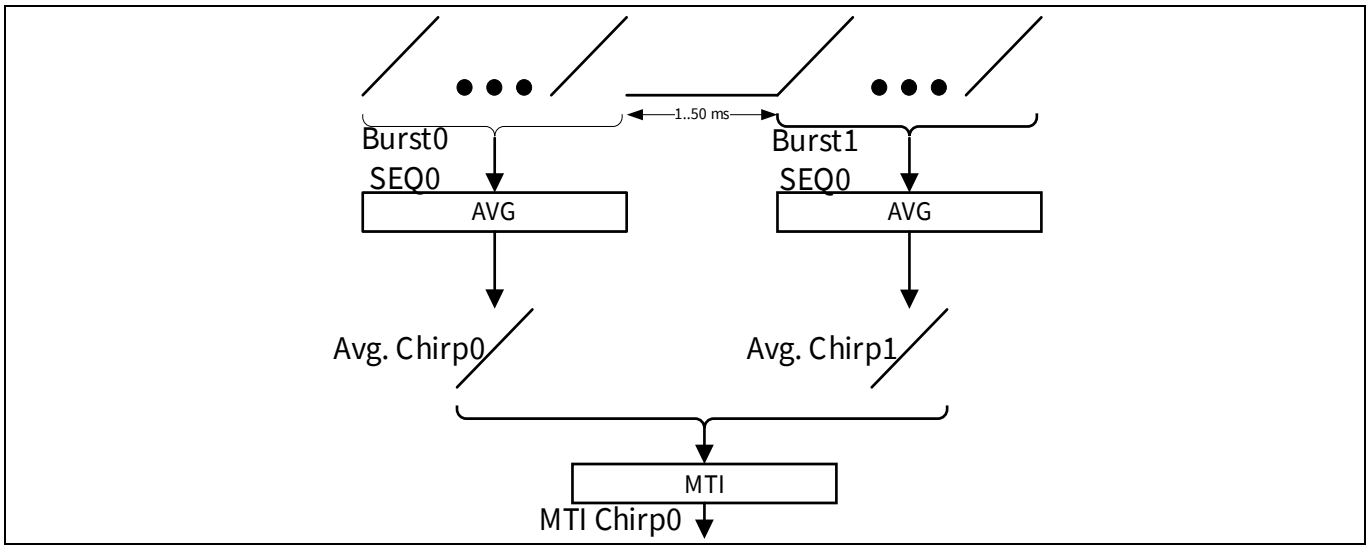


Figure 71 MTI delta mode

7.1.3.2 Windowing

To minimize the effect of spectral leakage of the FFT, there is the option to apply a window function before performing the FFT. The window scaling values of each sample are calculated based on the FFT size (number of samples) and the chosen window type. The optional window feature can be enabled via `MD_ULP_ALG_CTRL.WIN_EN` or `MD_RD_ALG_CTRL.WIN_EN` and requires enabled window generation or a preloaded window, see Section “Window generation”

7.1.3.2.1 Window generation

To avoid storing several window factors, the on-chip integrated CORDIC block is used to calculate the required factors on the fly. This can be done as each factor is the sum of multiple cosine multiplications, see Equation 7.

$$w(n) = \sum_{k=0}^{K} (-1)^k * a_k cos\left(\frac{2\pi kn}{N}\right), 0 \le n \le N$$

Equation 7 The general form of a window function w(n) using a cosine series

Where:

- a_k are the coefficients of the cosine terms that determine the specific type of window
- N is the number of samples (the length of the window minus 1, or $L = N + 1$)
- K is the order of the cosine series; for many standard windows, K is typically 1 or 2, but more complex windows might use higher-order terms

The cosine terms are calculated using CORDIC in “rotation mode”. The CORDIC algorithm inherently introduces a gain error due to the iterative nature of the computation. This error can be corrected by scaling the input by a factor of 0.678 (10 iterations). Therefore, the coefficients (a1, a2, a3 and a4), which are part of a cosine term, must be scaled accordingly: $a_1^* = \frac{a_1}{0.678}$. Typical window types and respective coefficients are shown in Table 21.

For the input parameters A*, the format Q0.15 (1 signed, 0 integer, 15 fractional) is used. All parameters (a0, a1, a2, a3 and a4) can be programmed via `MD_R/CFFT_WIN_CTRL0/1/2` registers and the bitfields `A0`, `A1`, `A2`, `A3` and `A4`. The resulting window coefficients w[n] are in the format Q0.23 (1 signed, 0 integer, 23 fractional).

Window generation can be enabled with: `MD_ULP_ALG_CTRL.WIN_GEN_EN` or `MD_RD_ALG_CTRL.WIN_GEN_EN`.

Table 21 Typical window types and respective parameters

Window type	a0	–a1 / 0.6780	a2 / 0.6780	–a3 / 0.6780	a4 / 0.6780
Hann	0.5	–0.7375	0	0	0
Hamming	0.53836	–0.6809	0	0	0
Blackman	0.42659	–0.7324	0.1133	0	0
Blackman Harris	0.35875	–0.7202	0.2084	–0.0172	0
Flattop	0.21557895	–0.6145	0.4089	–0.1233	0

7.1.3.2.2 Window preloading via software

There is also an option to pre-load the window coefficients by software via SPI. In such a scenario, the MD_BUFFER needs to be enabled (powered) permanently, which is achieved by setting `MD_BUFFER_CTRL.PM=1B` (ON). The 24-bit coefficients are stored in the Q0.23 format (1 signed, 0 integer, 23 fractional) in the MD_BUFFER, whereas the bits [23:0] (`SPI_CTRL.MSB_ALIGN_EN=0`) or the bits [31:8] (`SPI_CTRL.MSB_ALIGN_EN=1`) of the 32-bit SPI data payload is considered.

Due to point symmetry, only one half of the overall window coefficients for each dimension need to be stored (i.e., up to 256 for up to 512-point RFFT).

The start address of the window coefficients in the MD_BUFFER is 0xF000. Table 22 and Table 23 show the coefficient addresses for ULP and RD modes.

Table 22 Address overview for window coefficients one dimensional (x = sample)

Address	Coefficient
0xF000	x(0)
0xF004	x(1)
0xF008	x(2)
…	…
0xF3FC	x(255)

Table 23 Address overview for window coefficients two dimensional (x = sample, y = burst)

Address	Coefficient
0xF000	x(0)y(0)
0xF004	x(1)y(0)
0xF008	x(2)y(0)
…	…
…	x(0)y(1)
…	x(1)y(1)
…	…
0xF7FC	x(N_x/2-1)y(N_y/2-1)

Note: The window generation feature must be disabled in that case to prevent overriding the transmitted window.

7.1.3.3 Range – Real FFT (RFFT)

The embedded range FFT converts the time domain signal to a frequency domain signal. Its output signal represents the amount of energy per FFT bin, where each FFT bin corresponds to a certain range. As the chip architecture is designed in a way to provide in-phase data only, all ADC data is processed as real part and consequently the imaginary input of the FFT is considered to be zero. The complex FFT frequency spectrum includes also the negative frequencies, see Figure 72. The positive frequencies correspond to the bins 1 to N/2 and the negative frequencies correspond to the bins N/2 + 1 to N-1, where N is the number of input samples. Due to its conjugate complex symmetry, which means that for real input data, the real part of the complex FFT has even symmetry and the imaginary part has odd symmetry, and the upper half of the FFT result can be ignored, see Figure 73. Bin 0 covers the DC part which represents the offset of the input signal.

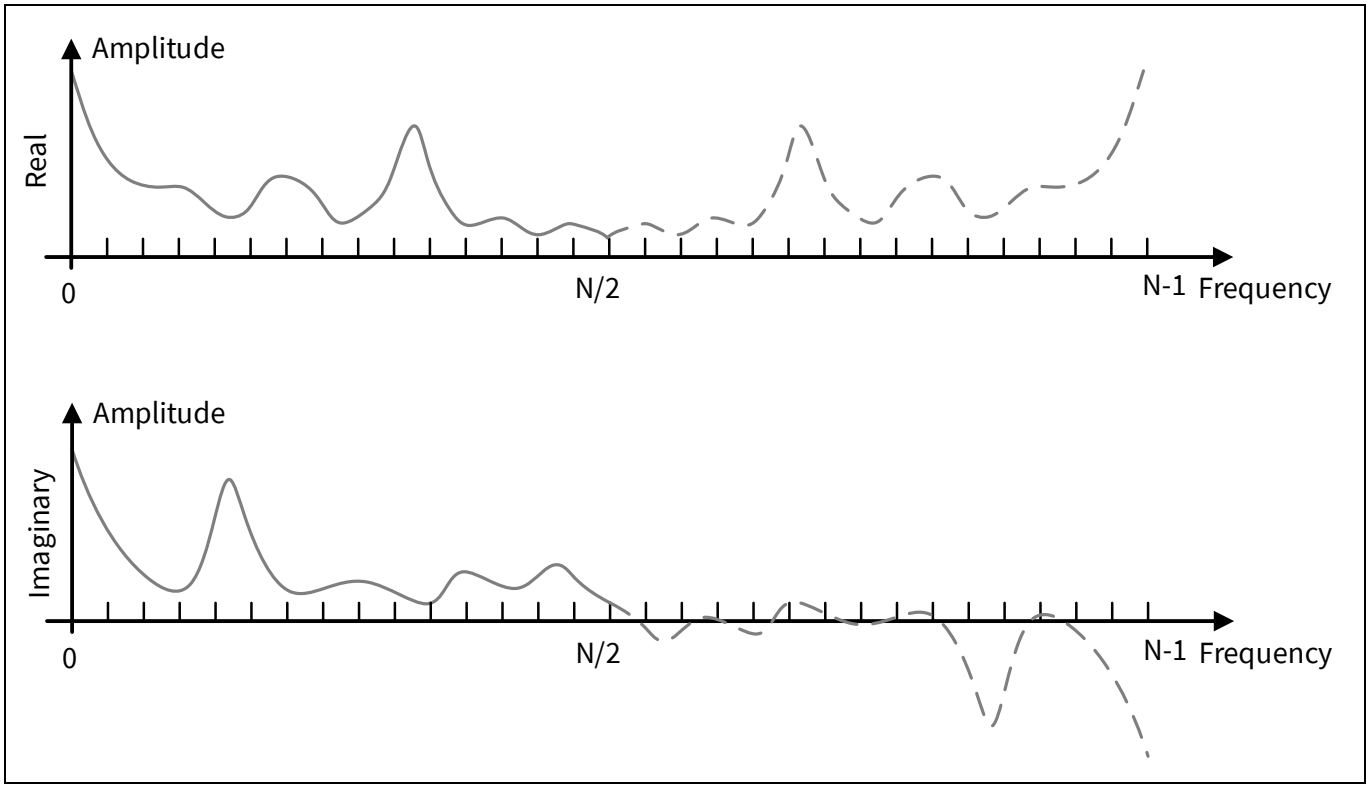


Figure 72 RFFT spectrum

Due to the previously described symmetry, only the lower half of the spectrum is used for further processing.

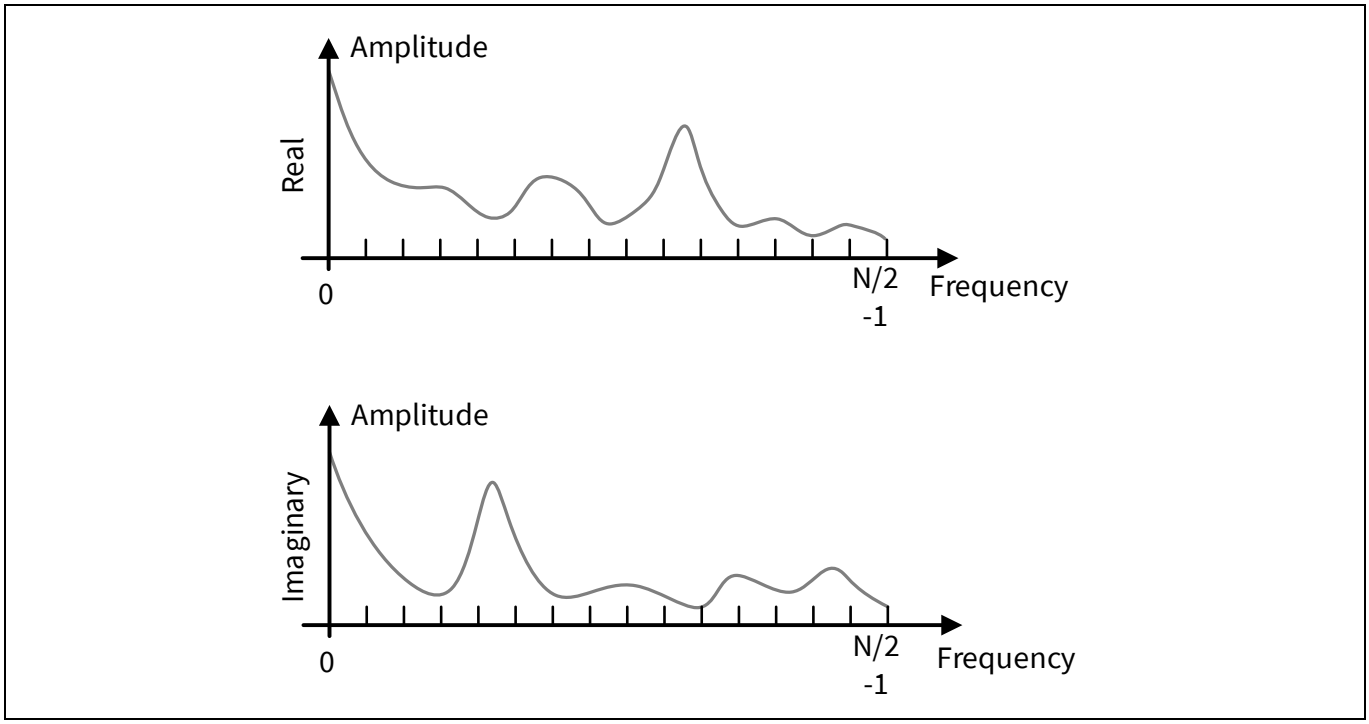


Figure 73 RFFT spectrum result

The embedded RFFT for RD supports lengths of the form 2^x with x = {4, 5, 6, 7, 8, 9} which is defined by the number of samples per chirp (see SH_IDX# and MD_RD_CTRL.RFFT_SIZE). The embedded RFFT for ULP supports lengths of the form 2^x with x = {6, 7, 8, 9} which is defined by the number of samples per chirp (see SH_IDX# and MD_ULP_CTRL.RFFT_SIZE) and based on the use-case scenario, the application has to choose between following trade-offs.

RFFT is enabled with `MD_ULP/RD_ALG_CTRL.RFFT_EN`.

7.1.3.4 Doppler – Complex FFT (CFFT)

After the range FFT (fast time) was applied on all chirps, the Doppler (complex) FFT is performed (slow time) across the averaged chirps in the RD operation mode. The Doppler FFT provides additional information about the velocity. The calculation process is shown in Figure 74.

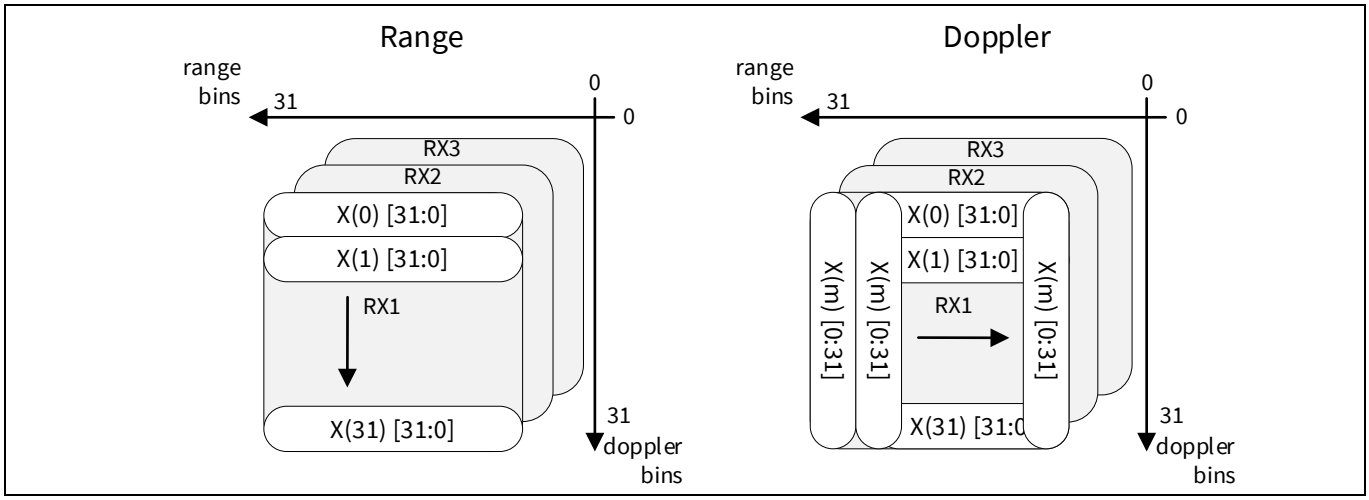


Figure 74 Doppler FFT processing

All processing of chirp averaging, range FFT and Doppler FFT is performed for each RX channel inside the RX_BUFFER memory. Due to the maximum depth of 1024 words in the memory, the number of range bins multiplied by the number of Doppler bins shall not exceed 1024 complex points. Table 24 shows valid number of range bins and number of Doppler bins for RD mode. The minimum and maximum Doppler FFT sizes are limited to 8 and 64 respectively.

Table 24 Valid combinations of FFT processing for RD mode

Valid combinations	Number of range bins = (number of samples)/2	Number of Doppler bins (number of avg. chirps without or with MTI enabled)
Option 1	8 (16)	8 (8 or 9)
Option 2	8 (16)	16 (16 or 17)
Option 3	8 (16)	32 (32 or 33)
Option 4	8 (16)	64 (64 or 65)
Option 5	16 (32)	8 (8 or 9)
Option 6	16 (32)	16 (16 or 17)
Option 7	16 (32)	32 (32 or 33)
Option 8	16 (32)	64 (64 or 65)
Option 9	32 (64)	8 (8 or 9)
Option 10	32 (64)	16 (16 or 17)
Option 11	32 (64)	32 (32 or 33)
Option 12	64 (128)	8 (8 or 9)
Option 13	64 (128)	16 (16 or 17)
Option 14	128 (256)	8 (8 or 9)

Note: For options where number of Doppler bins exceed 32, Doppler masking can be applied only up to first 32 Doppler bins. For number of Doppler bins exceeding 32, it is recommended to enable MTI to remove static targets.

The number of averaged chirps (bursts) must be configured for the Doppler FFT using the register `MD_RD_CTRL.NUM_BURST`, set to a value in the form of 2^x. If MTI is enabled, the value must be in the form $2^x + 1$. Also, `MD_RD_CTRL.CFFT_SIZE` must be configured to 2^x.

CFFT is enabled with `MD_RD_ALG_CTRL.CFFT_EN` and requires prior RFFT.

7.1.3.5 Coordinate rotation digital computer (CORDIC)

The coordinate rotation digital computer (CORDIC) block transforms the complex cartesian output data of the FFT to a polar representation with magnitude and phase.

The format is in Q0.23 (1 sign bit, 0 integer bit, 23 fractional bits) and covers the range from [-1, 1] Where –1 is included and +1 is excluded.

Note: The CORDIC module is not configurable.

7.1.3.6 Constant false alarm rate (CFAR)

Ordered statistic constant false alarm rate (OS-CFAR) is an adaptive threshold detection method, which enables target detection on varying environment conditions. It is applied to identify targets within a range- or a range-Doppler map.

CFAR is enabled with `MD_ULP_ALG_CTRL.CFAR_EN` or `MD_RD_ALG_CTRL.CFAR_EN` and requires prior execution of RFFT for ULP mode and CFFT for RD mode.

In OS-CFAR, each RFFT/CFFT bin gets tested for a proper target condition. The current cell under test (CUT) is compared against a calculated threshold, which is based on its neighboring cells, also called window (WIN) cells, a user configurable scaling factor, and an offset. If the CUT surpasses the calculated threshold, a target is identified and flagged.

Since power peaks occur not only in a single bin but also in adjacent bins, the reference window does not begin immediately next to the CUT. The cells adjacent to the CUT, known as guard cells (GUARD), are excluded from the threshold calculation. The size of the guard is adjustable.

Figure 75 shows a one-dimensional CFAR, with 4 WIN cells and 2 GUARD cells.

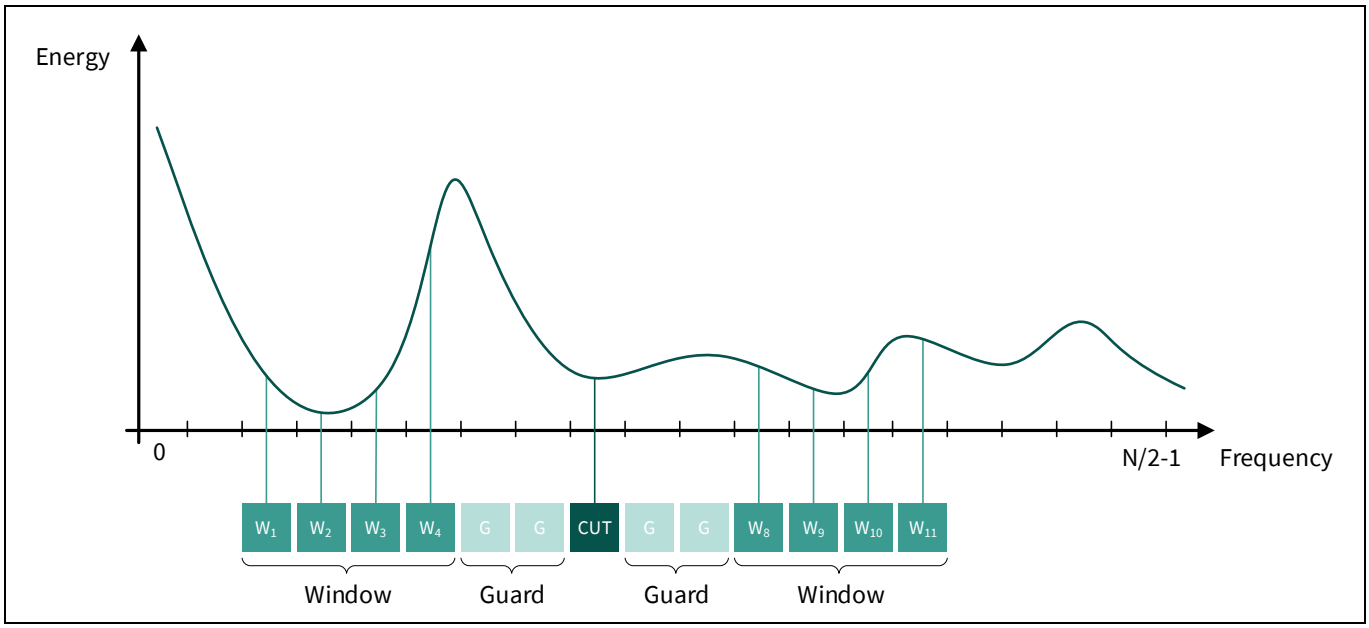


Figure 75 1D – CFAR cell selection

A two-dimensional OS-CFAR can be applied to Range Doppler maps to achieve a better estimation of the noise floor. The threshold calculation for a specific cell, known as the cell under test (CUT), depends on the activity in both neighboring range bins and neighboring Doppler bins. The number of window cells and number of guard cells in both range and Doppler directions are configurable.

Figure 76 shows a 2D OS-CFAR with 4 WIN cells and 1 GUARD cell in each of the range and Doppler directions. START and STOP for range and Doppler domain define the size of the range-Doppler map which is analyzed by the CFAR. Cells outside of the defined start and stop bin are ignored by the CFAR processing. In contrast to that, the exclude (EXCL) bitfields for range and Doppler are analyzed by the CFAR and reported in the CFAR binary map. However, targets which fell inside such exclude range or Doppler bins are never reported in the target registers (`MD_TARGET`).

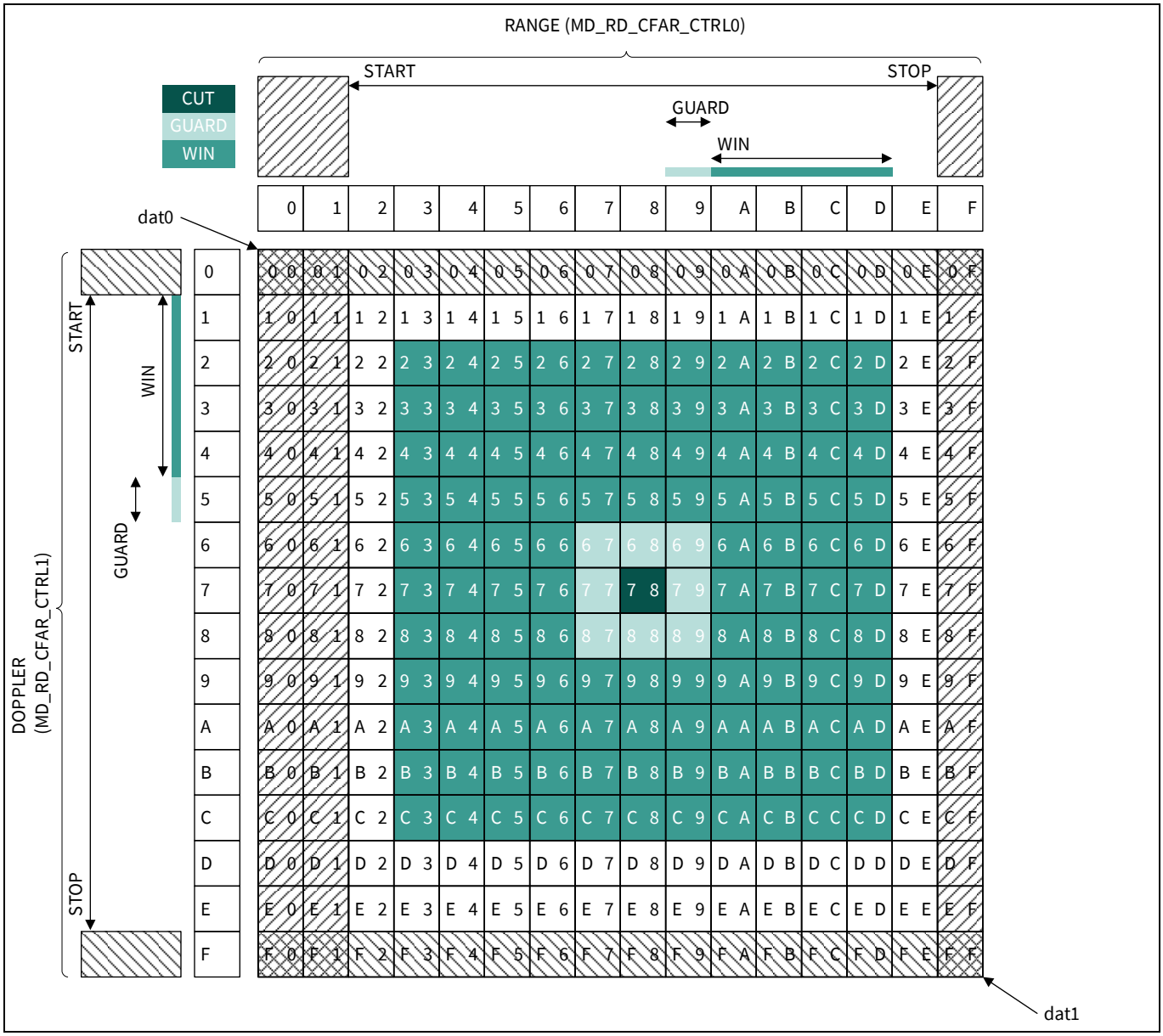


Figure 76 2D OS-CFAR

An OS-CFAR iteration begins with sorting all window cells in ascending order based on their magnitude. Once completed, a certain cell k is selected as reference. The cell k is chosen as the k-th quantile of active window cells, see Equation 8 and Figure 77.

$$k = N_{WIN} \times K_{SEL}$$

Equation 8 k-th quantile of OS-CFAR

Where N_{WIN} is the number of active window cells. Active window cells are all window cells with valid FFT data. If the CUT is close to the corner of an FFT array, such that a part of the window exceeds the FFT array size, k is automatically adjusted by the reduction of active window cells (N_{WIN}).

K_SEL is defined as fixed-point value in the Q0.7 format, representing a range from 0 (0%) to +0.9999 (~100%). Only values in the interval [0;1] are allowed.

For example, if K_SEL is 0.25 and there are 8 active window cells, then 2nd smallest window cell is used as reference to estimate the noise level.

The k-th cell of the window cells is finally multiplied with a scaling factor and added to an offset value before it gets compared to the value of the cell under test to determine whether a target is present.

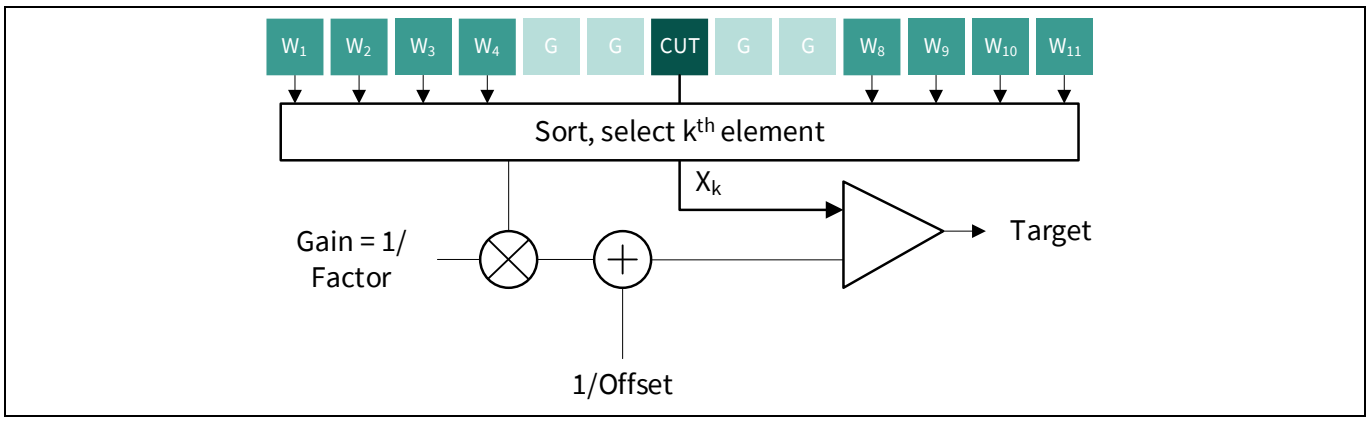


Figure 77 OS-CFAR

There is an option to limit the CFAR operation to a certain range or Doppler region if not all range bins/Doppler bins are of interest. This can be done by defining a start (START) and stop (STOP) limit for the CFAR operation, see Figure 76.

Note: The range during CFAR operation is limited to 254. Therefore, the STOP bitfield must not be set to 255 (0xFF).

All parameters mentioned in Section Constant false alarm rate (CFAR) are configurable and can be defined via the register `MD_ULP_CFAR*` and `MD_RD_CFAR*` based on the application requirements.

The overall number of range bins is divided into four range zones. Each range zone comes with its own scaling factor in the format Q0.15 (1 signed, 0 integer bit, 15 fractional bits). The scaling factor is defined in the format gain = 1/factor. This allows different sensitivities for detecting targets depending on the distance from the sensor (see Figure 78).

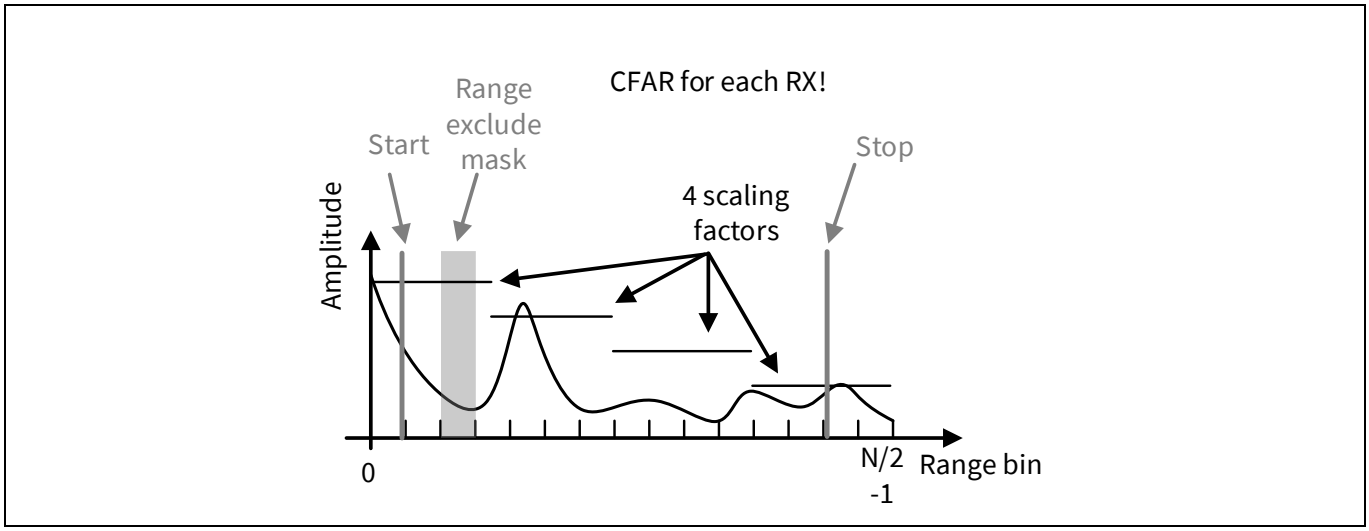


Figure 78 OS-CFAR scaling

Figure 79 shows an example of a CFAR result with 12 identified targets. There are 16 averaged chirps (bursts), each containing 32 samples resulting in 16 complex range bins and 16 complex Doppler bins. For range, the start of the CFAR processing is configured to 2, so that the first two range bins are ignored. The range stop is configured to 15, ensuring that the last range bin is also excluded from processing.

For Doppler processing, the start and stop values are configured, such that the first and last Doppler bins, representing the minimum (–Vmin) and maximum velocity (+Vmax) are also excluded from the processing.

Note: Applied range and Doppler exclude bins are not considered at this step and are evaluated during the target extraction step.

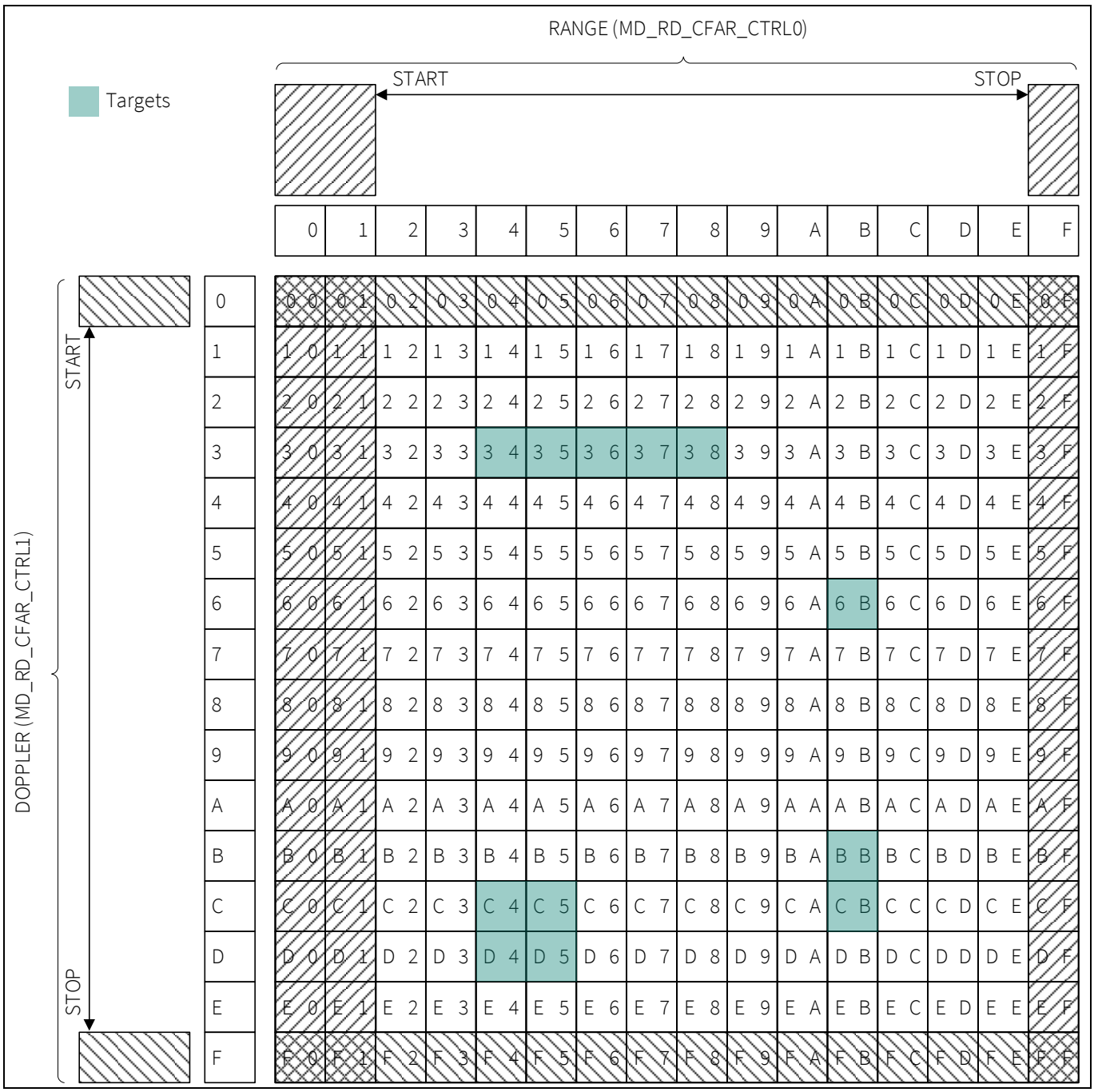


Figure 79 Example of 2D OS-CFAR

7.1.3.6.1 Constant threshold operation

The CFAR module can be configured to act as a constant threshold instead of using an adaptive noise estimation for target detection. A constant threshold is achieved by programming the desired value into `MD_RD_CFAR_OFFSET` or `MD_ULP_CFAR_OFFSET` and disabling the noise window as shown in the code listing below for an RD sequence.

Note: The offset register holds 32 bit, but the internal datapath considers only the most significant 24 bit.

Code listing 10 Constant threshold operation

```
MD_RD_CFAR_OFFSET.VAL=t // constant threshold t for CFAR
MD_RD_CFAR_CTRL0.K_SEL=0x7F // k_sel to almost 1.0
MD_RD_CFAR_CTRL0.WIN=0 // Disable CFAR window
MD_RD_CFAR_CTRL1.WIN=0 // Disable CFAR window
MD_RD_CFAR_CTRL0.GUARD=0 // Disable CFAR guard cells
MD_RD_CFAR_CTRL1.GUARD=0 // Disable CFAR guard cells
MD_RD_CFAR_SCALE_L.FACTOR0=0x7FFF // CFAR gain to almost 1.0
MD_RD_CFAR_SCALE_L.FACTOR1=0x7FFF // CFAR gain to almost 1.0
MD_RD_CFAR_SCALE_H.FACTOR2=0x7FFF // CFAR gain to almost 1.0
MD_RD_CFAR_SCALE_H.FACTOR3=0x7FFF // CFAR gain to almost 1.0
```

7.1.3.7 Merge

Boolean merge (merge) is used to reduce multiple CFAR result maps into one single map. The reduction is performed based on one of the following options:

- **Boolean OR-combination:** A target is kept if at least one RX channel identifies a target at a certain bin (*MD_ULP/RD_CFAR_CTRL1.RX_MERGE_SEL=0*)
- **Boolean AND-combination:** A target is kept if all RX channels identify a target with the same range bin and same Doppler bin (*MD_ULP/RD_CFAR_CTRL1.RX_MERGE_SEL=1*)

If only one RX antenna is active, executing merge has no effect on the output. Merge is recommended to be enabled when multiple RX antennas are active. Merge can be enabled with `MD_ULP/RD_ALG_CTRL.RX_MERGE_EN`.

7.1.3.8 Local maximum

Local maximum detection enables precise peak detection when multiple adjacent bins are marked as valid targets due to the presence of a single target. This approach ensures that the single targets are not reported multiple times, thereby enhancing the accuracy of Angle extraction.

The target map to be refined is the target list of either the original output of Constant false alarm rate (CFAR) or the output based on the Merge. If merge is enabled, all enabled antennas are accumulated together by a non-coherent sum shown in Equation 9.

$$\text{Magnitude}_{\text{Target}} = \text{Magnitude}_{\text{CUT_RX1}} + \text{Magnitude}_{\text{CUT_RX2}} + \text{Magnitude}_{\text{CUT_RX3}}$$

Equation 9 Local maximum: non-coherent sum

If Merge is disabled, a local maximum is extracted for each antenna individually. Local maximum operation is performed on CFFT data in the case of range Doppler mode, or on RFFT data in the case of ULP mode. This operation identifies those cells as main target points where the mean magnitude value across all active receiver channels is higher than that of their local neighboring cells. In the example shown in Figure 80, the local maximum refines the target list such that only 2 targets are kept.

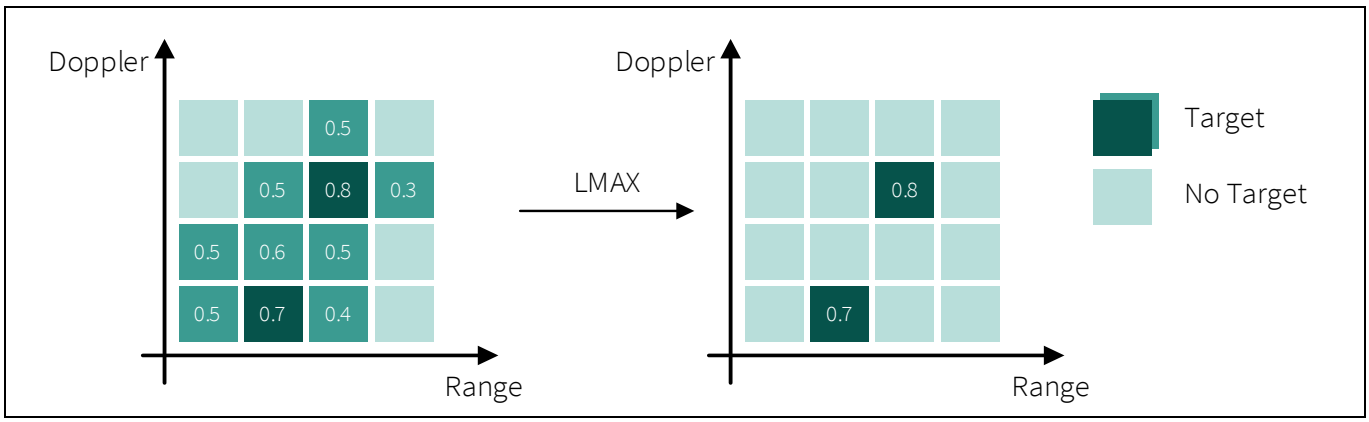


Figure 80 Local maximum extraction of targets

The local maximum is enabled with `MD_ULP/RD_ALG_CTRL.LOC_MAX_EN` and requires prior execution of Constant false alarm rate (CFAR).

7.1.3.9 Target extraction

The target extraction step extracts a list of remaining detections after Local maximum or Constant false alarm rate (CFAR) and extracts the following properties. These properties include the location (range bin, Doppler bin, number of RX antenna), the magnitude, and optionally the angle of arrival for both H-plane and E-plane.

Up to eight detections will be listed. All detections listed will be ordered with respect to range (from close to far).

Each detection register set consists of three registers with the following information:

- **MD_TARGET#_MAG:** Magnitude of detection from the RX channel with the highest magnitude in Q0.31 format (1 sign bit, 0 integer bit, 31 fractional bits). An additional scaling factor of 1.475 (=1/0.677954) is applied to the result
- **MD_TARGET#_LOC:** Range bin, Doppler bin, and the number of RX antenna
- **MD_TARGET#_ANGLE:** H-plane and E-plane angle information in Q0.15 format (1 sign bit, 0 integer bit, 15 fractional bits)

Note: Target extraction is enabled with `MD_ULP_ALG_CTRL.TARGET_EN` or `MD_RD_ALG_CTRL.TARGET_EN` and requires CFAR to be enabled. All detection registers are cleared at the beginning of the processing to ensure they do not contain outdated target data from previous frames.

The number of actual targets detected within the field of view depends on the level of environmental clutter as well as the FMCW chirp ramp configuration. To have real detections of multiple targets, the targets have to be separated in range or doppler or angular domain. The observed detections could also originate from a single target, depending on its radar cross section (RCS).

7.1.3.9.1 Angle extraction

Angle extraction is a sub step of Target extraction and can be used to extract an angle of arrival (AoA) for H-plane and E-plane. It can be enabled for ULP as well as for RD via `ULP_ALG_CTRL.ANGLE_EN` or `RD_ALG_CTRL.ANGLE_EN`.

If enabled, angle extraction can be used to further refine the field of view by selectively masking certain angles (see Angle masking) or certain range angle zones (Range-angle zone masking), which may be irrelevant for the application.

The angle extraction itself is subdivided into calibration and calculation and is executed for H-Plane as well as for E-plane for each target, as shown in Figure 81.

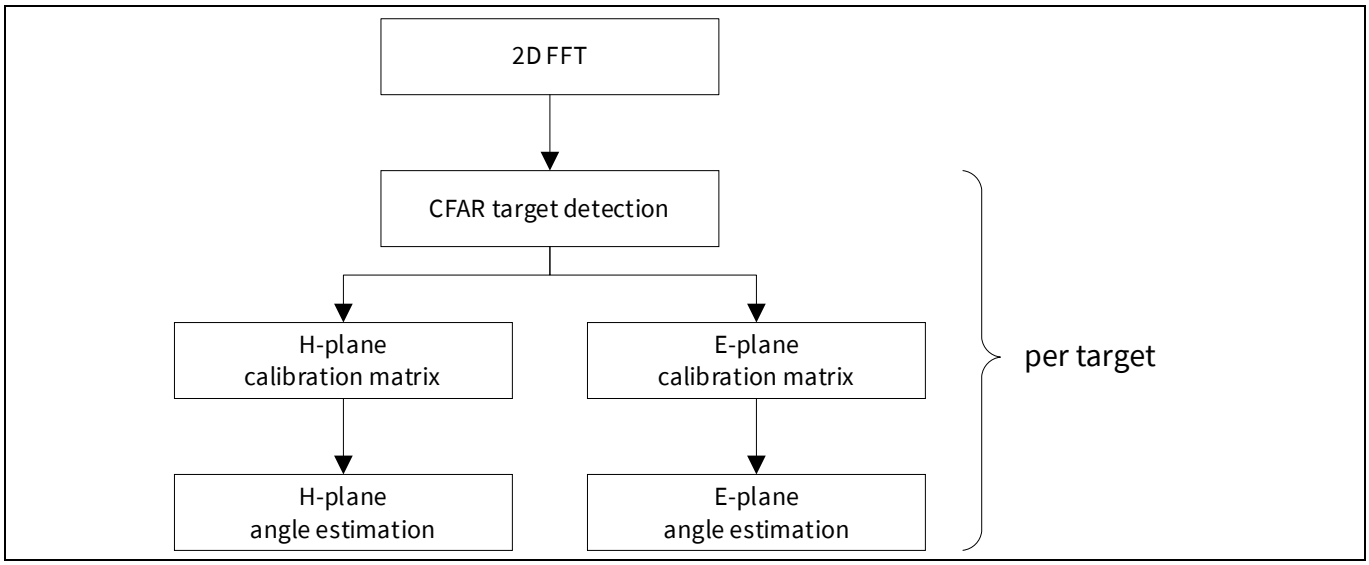


Figure 81 Angle extraction processing

7.1.3.9.2 Calibration

Imperfections from the transmitter (TX) to receiver (RX) path on antenna side, especially for antenna in package, have a non-negligible impact on the phase information. Furthermore, reflections coming from very short distances like housings or covers of the radar sensor have an influence on the phase. To compensate for such impacts, it is inevitable to apply a calibration matrix prior to the angle calculation. The calibration parameters are specific to the hardware and are provided by Infineon.

Calibration matrix in E- and H-plane:

Two calibration matrices are available, typically one for H-Plane (`MD_CAL_MAT_H**`) and one for E-plane (`MD_CAL_MAT_E**`). See Equation 10. A typical calibration matrix with complex numbers generated from MATLAB is shown in Section “Data post-processing”.

$$\boldsymbol{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix}$$
$$\boldsymbol{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{31} & E_{32} & E_{33} \end{bmatrix}$$

Equation 10 Calibration matrices in H- and E-plane with complex parameters H_{xy} and E_{xy}

The complex calibration matrix parameters H_{xy} ($x = [1,2,3]$ and $y = [1,2,3]$) can be formed out of the register bit fields MD_CAL_MAT_H_xy.A and MD_CAL_MAT_H_xy.B which represents the real and imaginary part in the Q0.15 format (1 sign bit, 0 integer bit, 15 fractional bits). Equation 11 shows how to form the complex calibration matrix parameters out of the register values.

$$H_{xy} = \text{MD_CAL_MAT_H_xy.}A + j \cdot \text{MD_CAL_MAT_H_xy.}B$$
$$E_{xy} = \text{MD_CAL_MAT_E_xy.}A + j \cdot \text{MD_CAL_MAT_E_xy.}B$$

Equation 11 Formulation of complex calibration parameters out of register values

Condition for calibration matrix coefficients:

To avoid signal saturation during MMIC internal processing, the sum of the absolute values of the H-plane and E-plane coefficients (as shown in Equation 12) must not exceed 1.0 or less.

$$\begin{aligned} abs(H_{11}) + abs(H_{12}) + abs(H_{13}) &\leq 1 \\ abs(H_{21}) + abs(H_{22}) + abs(H_{23}) &\leq 1 \\ abs(E_{11}) + abs(E_{12}) + abs(E_{13}) &\leq 1 \\ abs(E_{31}) + abs(E_{32}) + abs(E_{33}) &\leq 1 \end{aligned}$$

Equation 12 Condition for calibration matrix coefficients in both, E-plane and H-plane

Obtain complex receiver antenna signals:

- The receiver antenna signals are complex valued, resulting from fast Fourier transform (FFT) processing during previous steps
- These signals are obtained from the range FFT result or range Doppler (RD) map, where signals from the same range and Doppler bin are selected to form the receiver antenna array $[RX_1 \quad RX_2 \quad RX_3]'$

Apply calibration matrix:

Next, multiply the complex valued receiver antenna array (3 x 1) by the calibration matrix (2 x 3). This will produce the calibrated signals $A_{H/Eplane}$ and $B_{H/Eplane}$ as shown in Equation 13 and Equation 14.

$$\begin{bmatrix} A_{Hplane} \\ B_{Hplane} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \end{bmatrix} \cdot \begin{bmatrix} RX_1 \\ RX_2 \\ RX_3 \end{bmatrix}$$

Equation 13 H-plane calibration matrix applied to the complex-valued signals from each RX antenna after FFT

$$\begin{bmatrix} A_{Eplane} \\ B_{Eplane} \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \cdot \begin{bmatrix} RX_1 \\ RX_2 \\ RX_3 \end{bmatrix}$$

Equation 14 E-plane calibration matrix applied to the complex valued signals from each RX antenna after FFT

Extracting phase information after calibration matrix was applied:

After applying the calibration matrix, the phase differences $\Delta\alpha_{Hplane}$ and $\Delta\alpha_{Eplane}$ between (A_{Hplane}, B_{Hplane}) and (A_{Eplane}, B_{Eplane}) are extracted and wrapped to π, see Equation 15.

$$\Delta\alpha_{Hplane} = \text{wrap2pi}\left(B_{Hplane} - A_{Hplane}\right)$$
$$\Delta\alpha_{Eplane} = \text{wrap2pi}\left(B_{Eplane} - A_{Eplane}\right)$$

Equation 15 Phase differences and phase wrapping

Optimizing power consumption for single-plane angle measurements:

If your application only requires measurement of either the H-plane or E-plane angle, you can optimize power consumption by disabling the unused RD channel. To do this:

1. Set the unused RD channel to zero by configuring the CS_IDX#.RX#_EN register to 0
2. Disable the unused RD channel in the data processing stage by setting the MD_RD_CTRL.RX#_ACT or MD_ULP_CTRL.RX#_ACT register to 0

Note: When disabling an RD channel, you must also set the corresponding calibration matrix to zero for the respective plane. This will result in an angle measurement of approximately 0° for the unused plane.

7.1.3.9.3 Angle calculation

The angle calculation is based on a phase comparison monopulse (PCM) concept. The basic principle of PCM involves using two receiver antennas spaced by a certain distance apart. When a radar wave is reflected by a target and received by antennas, the difference in the path lengths to each antenna results in a phase difference between the received signals. This phase difference can be used to calculate the angle of arrival (AoA) with Equation 16.

$$\varphi_{\text{H}plane} = \arcsin\left(\frac{\lambda \times \Delta\alpha_{\text{H}plane}}{4 \times \Delta\text{distance}_{\text{rx}_{\text{rx}}}}\right); \varphi_{Eplane} = \arcsin\left(\frac{\lambda \times \Delta\alpha_{Eplane}}{4 \times \Delta\text{distance}_{\text{rx}_{\text{rx}}}}\right)$$

Equation 16 Angle calculation

Where λ represents the wavelength of the used frequency and $\Delta\alpha_{\text{H}plane}$, $\Delta\alpha_{\text{E}plane}$ represents the calculated phase differences between the desired two receiver antennas of the identified target.

Since the wavelength and the distance of the receiver antennas is constant for the application, you need to set the value (λ/4xΔdistance_rx_rx) via the register `MD_ANGLE_CTRL1.ARCSIN_H` and `MD_ANGLE_CTRL1.ARCSIN_E`.
Factor ¼ in the above equation is used to scale the constant (λ/ Δdistance_rx_rx) to fit to the register, which is in the format Q0.23 (1 sign bit, 0 integer bit, 23 fractional bits) for internal processing.

The result of arcsine angle can be read from the target result register `MD_TARGETx_ANGLE` in Q0.15 format (1 sign bit, 0 integer bit, 15 fractional bits), mapping a value in the range of 2 radians $[-2\pi, +2\pi[$. To extract the angle δ_H, δ_E in degrees, the result needs to be multiplied with 360. See Equation 17

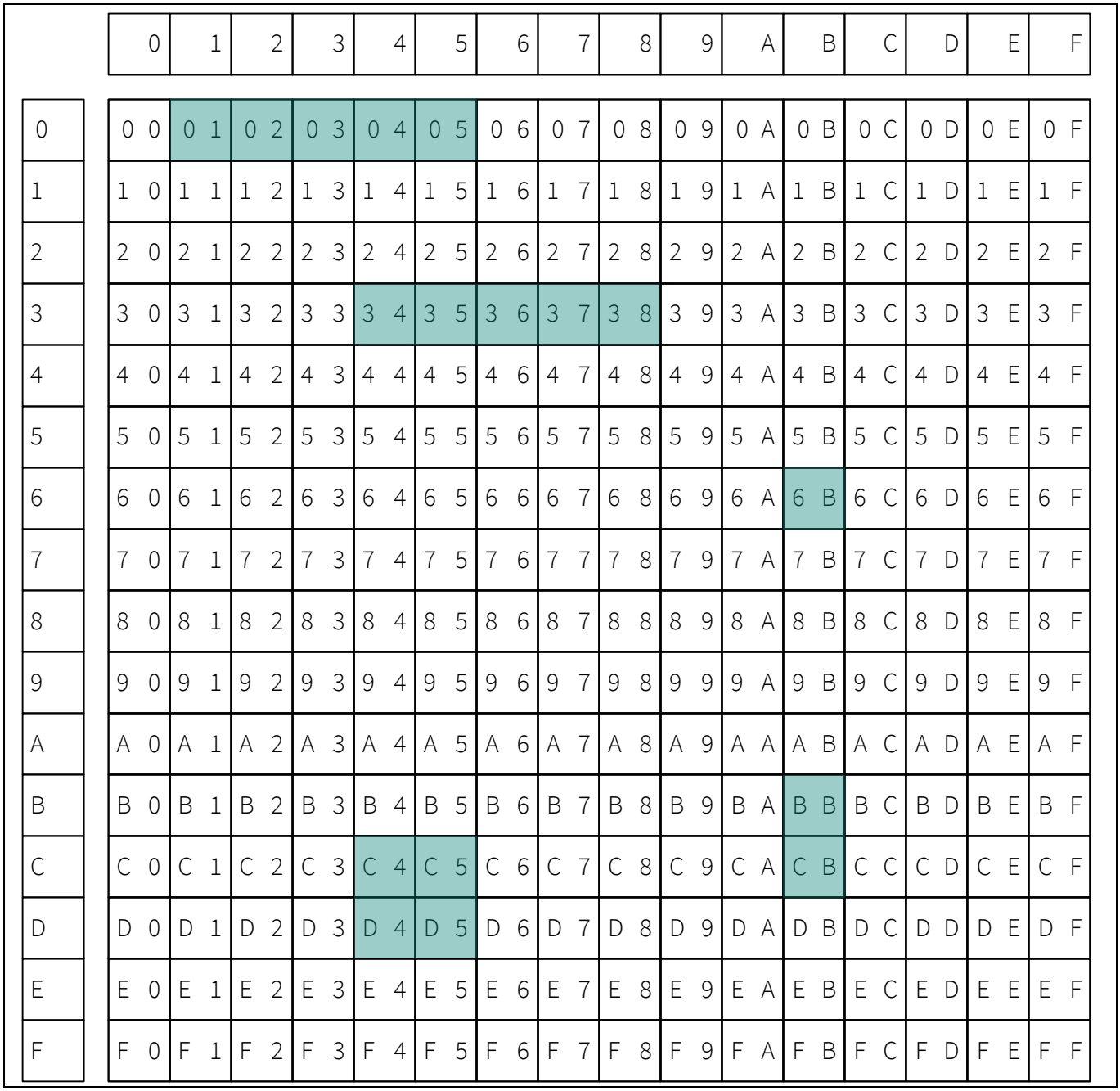
$$\delta_H = \text{MD_TARGETx_ANGLE.H_PLANE} \cdot 360°;\ \delta_E = \text{MD_TARGETx_ANGLE.E_PLANE} \cdot 360°$$

Equation 17 Angle extraction in degrees

7.1.3.10 Masking

Multiple masking stages can be used to select which targets get reported.

The following example shows different masking stages. Figure 82 shows an RD frame with complex values and potential targets after CFFT algorithm.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 0	0 1	0 2	0 3	0 4	0 5	0 6	0 7	0 8	0 9	0 A	0 B	0 C	0 D	0 E	0 F
1	1 0	1 1	1 2	1 3	1 4	1 5	1 6	1 7	1 8	1 9	1 A	1 B	1 C	1 D	1 E	1 F
2	2 0	2 1	2 2	2 3	2 4	2 5	2 6	2 7	2 8	2 9	2 A	2 B	2 C	2 D	2 E	2 F
3	3 0	3 1	3 2	3 3	3 4	3 5	3 6	3 7	3 8	3 9	3 A	3 B	3 C	3 D	3 E	3 F
4	4 0	4 1	4 2	4 3	4 4	4 5	4 6	4 7	4 8	4 9	4 A	4 B	4 C	4 D	4 E	4 F
5	5 0	5 1	5 2	5 3	5 4	5 5	5 6	5 7	5 8	5 9	5 A	5 B	5 C	5 D	5 E	5 F
6	6 0	6 1	6 2	6 3	6 4	6 5	6 6	6 7	6 8	6 9	6 A	6 B	6 C	6 D	6 E	6 F
7	7 0	7 1	7 2	7 3	7 4	7 5	7 6	7 7	7 8	7 9	7 A	7 B	7 C	7 D	7 E	7 F
8	8 0	8 1	8 2	8 3	8 4	8 5	8 6	8 7	8 8	8 9	8 A	8 B	8 C	8 D	8 E	8 F
9	9 0	9 1	9 2	9 3	9 4	9 5	9 6	9 7	9 8	9 9	9 A	9 B	9 C	9 D	9 E	9 F
A	A 0	A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A A	A B	A C	A D	A E	A F
B	B 0	B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B A	B B	B C	B D	B E	B F
C	C 0	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C A	C B	C C	C D	C E	C F
D	D 0	D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D A	D B	D C	D D	D E	D F
E	E 0	E 1	E 2	E 3	E 4	E 5	E 6	E 7	E 8	E 9	E A	E B	E C	E D	E E	E F
F	F 0	F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9	F A	F B	F C	F D	F E	F F

Figure 82 RD map with complex values and potential targets after CFFT

CFAR and merge algorithm will only consider targets inside the [start; stop] bitfields. Figure 83 shows the refined target map where targets in the first row are removed.

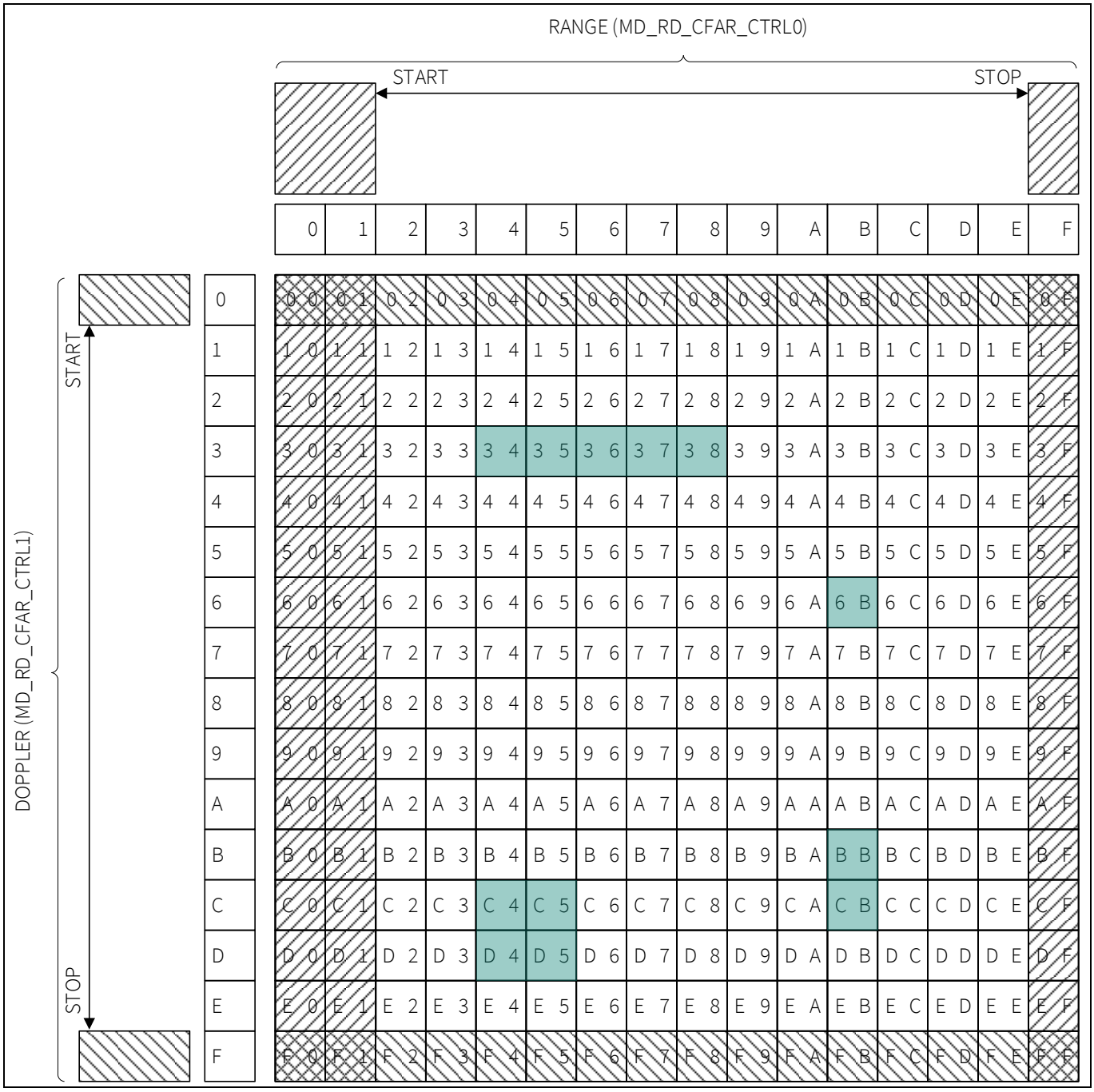


Figure 83 RD map with Boolean values and potential targets after CFAR & MERGE

7.1.3.10.1 Range masking

Range masking can be used to exclude targets which have a certain range. This can be a selective range (such as central RBIN) or a complete region (such as from central to last RBIN) or a combination of both.

Range bins can be excluded via `MD_ULP_RFFT_EXCL0`, `MD_ULP_RFFT_EXCL1`, `MD_RD_RFFT_EXCL0`, `MD_RD_RFFT_EXCL1` for ULP and RD, respectively. Each bit masks a range, which is defined by the chirp bandwidth and the number of samples per chirp. Register `EXCL0` and `EXCL1` bitfields are used in concatenation. In case more than 64 range bins are used, every bit from `EXCL` is duplicated. For 128 range bins, bit 0 is used for RBINS 0 and 1, bit 1 for RBINS 2 and 3. For 256 range bins, bit 0 is used for RBINS 0, 1, 2 and 3, bit 1 is used for RBINS 4, 5, 6 and 7.

7.1.3.10.2 Doppler masking

Doppler masking can be used to exclude targets which have a certain velocity. This can be a selective Doppler bin (such as central DBIN) or a complete region (such as from central DBIN to last DBIN) or a combination of both.

Doppler bins can be excluded via `MD_RD_CFFT_EXCL`, respectively. Each bit masks a velocity range, which is defined by the number of bursts and the burst repetition. Doppler Masking is available for configurations with up to 32 Doppler bins.

Both range and Doppler masking are applied during local maximum and target extraction algorithms. Masked bins are skipped for target evaluation but will still affect neighboring cells during local maximum search. Bins can get masked by the excluded neighboring cells. Lighter colors in Figure 84 show bins with smaller magnitude than with darker color. After local maximum, targets masked by range or Doppler exclude bins are not present in the target map anymore.

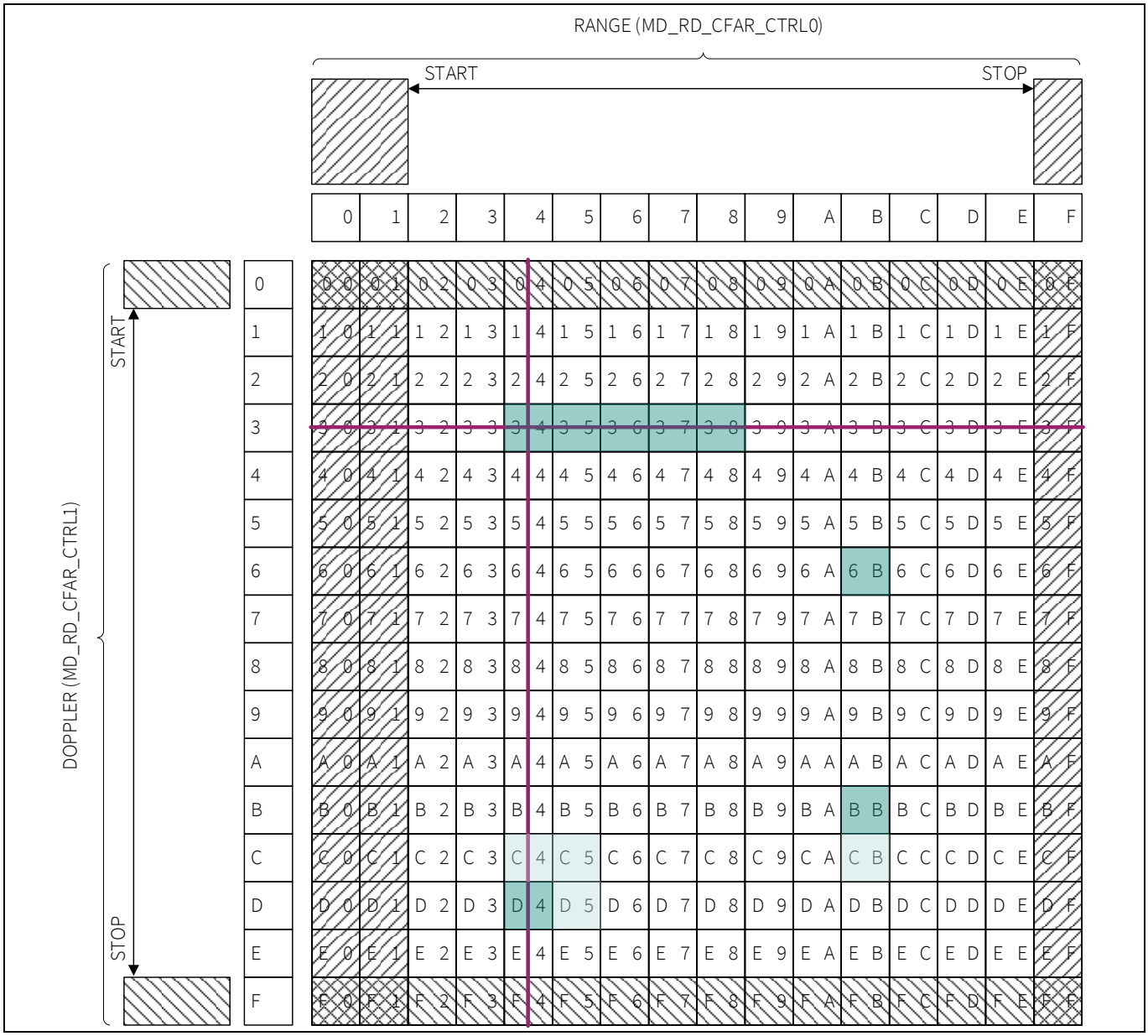


Figure 84 RD map with Boolean values during local maximum with range and Doppler exclude bits

7.1.3.10.3 Angle masking

Angle masking can be used to exclude targets which have a certain angle of arrival (AoA). This feature allows for the selective exclusion of specific regions (such as boresight) or a specific field of view (FoV) (such as from the horizon upwards), or a combination of both.

The H-plane and E-plane angles can be excluded independently via `MD_ANGLE_EXCL_H` and `MD_ANGLE_EXCL_E`, respectively. Each bit masks an angular range of ±2.8125° (=5.625°) and is mapped according to Figure 85.

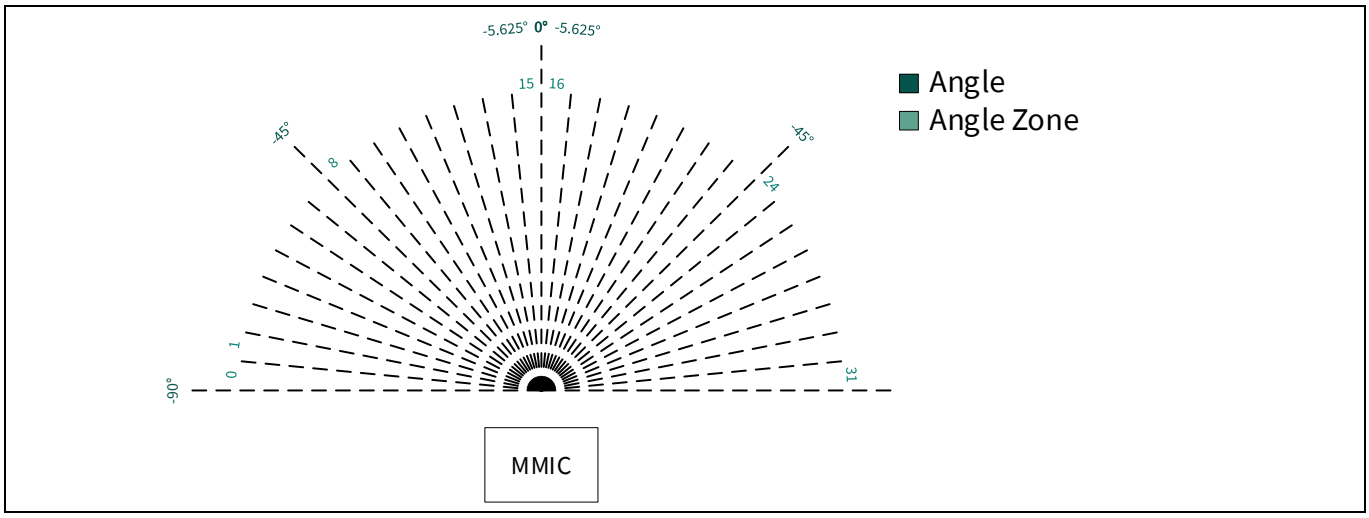


Figure 85 Angle filtering

Targets which fall into an excluded angle are skipped by the target extract step and will not be listed in the target registers (`MD_TARGET#`).

The previously refined target map is filtered once more with angle or range angle zone masking, as depicted in Figure 86.

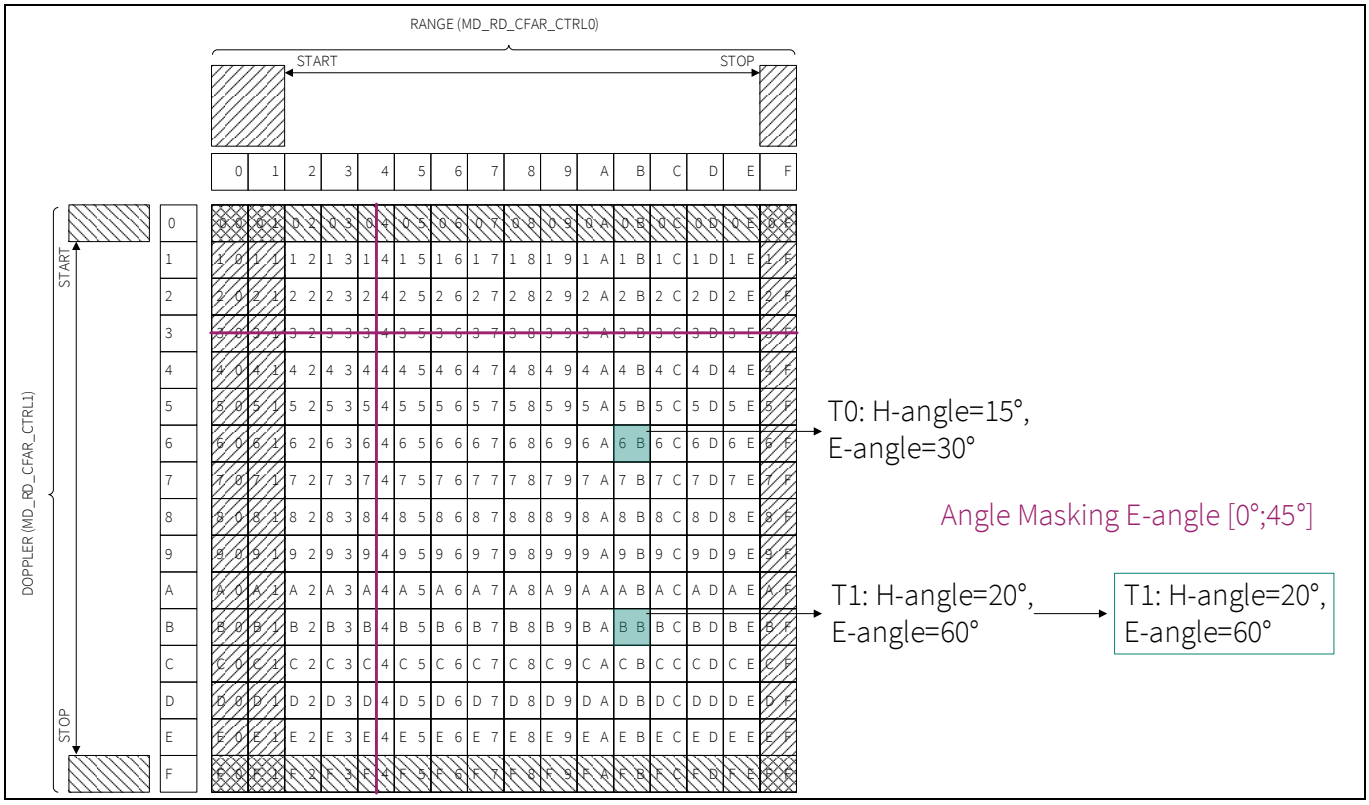


Figure 86 RD map with Boolean values during target extraction

7.1.3.10.4 Range-angle zone masking

Range-angle zone masking can be used to mask specific zones associated with particular combinations of range and angle, which cannot be masked by excluding the complete range or the complete angle.

Range-angle zones (RAZ) are clustered into seven angle zones and sixteen range zones, resulting in 112 RAZ polygons depicted in Figure 87 and Table 25. Each of these zones can be excluded individually. RAZ masking can be enabled either for H-Plane or the E-plane via `MD_ZONE_CTRL.PLANE_SEL`. The number of range bins per range-angle zone is not fixed but depends on the number samples per chirp (NUM_RANGE_BINS_RAZ = NUM_SAMPLE_PER_CHIRP / 2 / 16). Table 25 shows an example of interpreting sixteen range zones in scenarios involving both 32 range bins and 64 range bins. To obtain range zone resolution value in meters, the range resolution of respective sequence must be multiplied by the RANGE_WIDTH, as specified in the Equation 18.

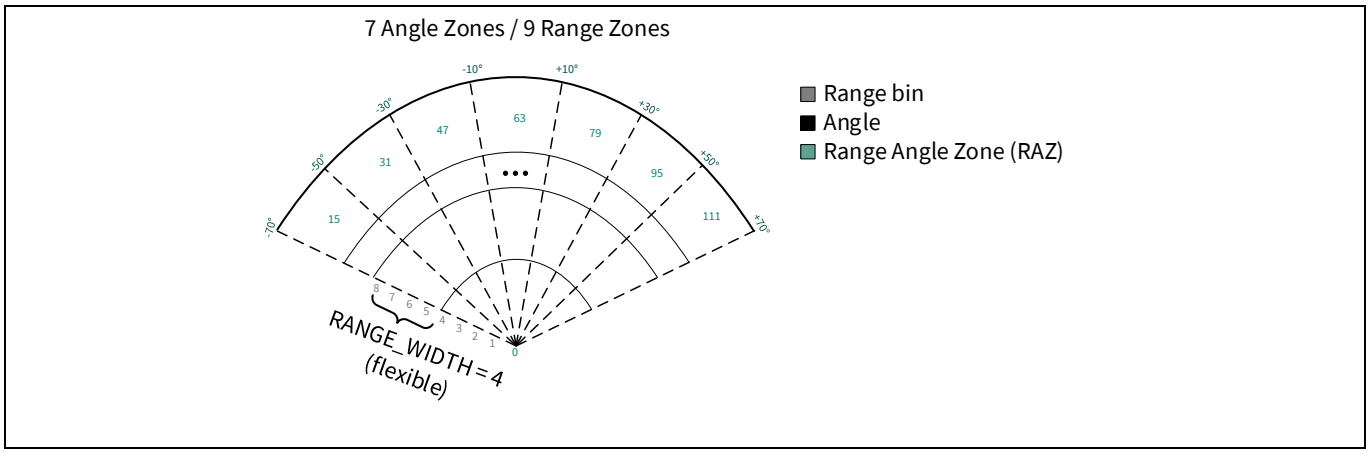


Figure 87 Range angle zones

$$ange\ zone\ resolution\ (RANGE_WIDTH) = \left[\frac{Number\ of\ Range\ bins}{16}\right]$$
$$Range\ zone\ resolution\ in\ meter\ = Range\ resolution\ in\ meter \times \left[\frac{Number\ of\ Range\ bins}{16}\right]$$

Equation 18 Range zone resolution

Table 25 Range zones

Range zone	16 range bins	32 range bins	64 range bins
0	0	0-1	0-3
1	1	2-3	4-7
2	2	4-5	8-11
3	3	6-7	12-15
4	4	8-9	16-19
5	5	10-11	20-23
6	6	12-13	24-27
7	7	14-15	28-31
8	8	16-17	32-35
9	9	18-19	36-39
10	10	20-21	40-43
11	11	22-23	44-47
12	12	24-25	48-51
13	13	26-27	52-55
14	14	28-29	56-59
15	15	30-31	60-63

Targets which fall into an excluded range angle zone are skipped by the target extract step and will not be listed in the target registers (`MD_TARGET*`, `MD_ZONE_STAT*`).

Targets outside of the ±78.75° (78.75° - 90°) are stored in the target register (`MD_TARGET#`) if not masked out via `MD_ANGLE_EXCL*` registers. However, these targets are not forwarded to the range angle zone status register (`MD_ZONE_STAT#`) as these zones are only defined up to ±78.75°.

7.1.3.11 Event counter

The event counter is an optional, additional filter for balancing a trade-off between false-positive rate and reaction time of the system. Setting the threshold of the counter to 0_D will effectively bypass this block and generate an event (interrupt) for each sequence, where at least one target is detected and not discarded by filtering in the target extraction step.

The operation of this filter is as follows:

- Each event will increase the counter, while the absence of an event will decrease the counter
- The increment and decrement step sizes can be specified with `MD_ULP_EVENT.INC` or `MD_RD_EVENT.INC` and `MD_ULP_EVENT.DEC` or `MD_RD_EVENT.DEC`, respectively
- When the counter surpasses the threshold (`MD_ULP_EVENT.TH` or `MD_RD_EVENT.TH`), it will generate an interrupt

- An interrupt will also be generated each time a frame with target is detected and the counter is above the threshold
- A separate saturation value can be specified with `MD_ULP_EVENT.SAT` or `MD_RD_EVENT.SAT`

Figure 88 shows an example of ULP or RD motion event interrupt generation with threshold 4_D, saturation 6_D, increment 2_D, and decrement step size 4_D.

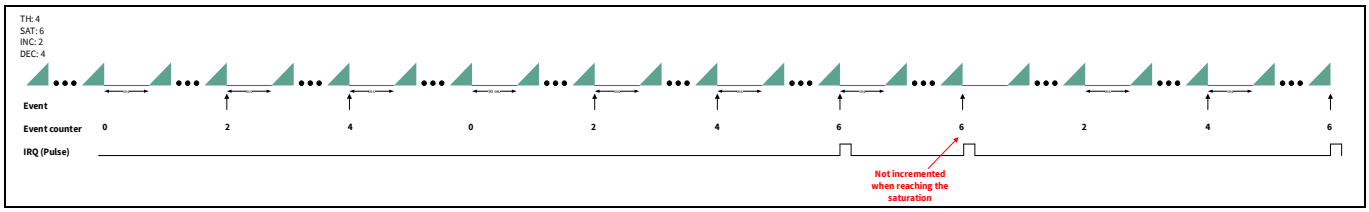


Figure 88 Motion event interrupt generation

To generate an interrupt on the digital I/O pin, the following need to be configured:

Code listing 11 ULP based motion event interrupt

```
MD_ULP_EVENT.TH=t // threshold t for event counter
MD_ULP_EVENT.SAT=s // saturation s for event counter
MD_ULP_EVENT.INC=i // increment step size i for event counter
MD_ULP_EVENT.DEC=d // decrement step size d for event counter
IR0.EN4=1B // enable ULP event at IR0
IO_DIO4.EN=1B // enable pin DIO4
IO_DIO4.MAP=4D // map pin DIO4 to IR0
```

Code listing 12 RD based motion event interrupt

```
MD_RD_EVENT.TH=t // threshold t for event counter
MD_RD_EVENT.SAT=s // saturation s for event counter
MD_RD_EVENT.INC=i // increment step size i for event counter
MD_RD_EVENT.DEC=d // decrement step size d for event counter
IR1.EN5=1B // enable RD event at IR1
IO_DIO5.EN=1B // enable pin DIO5
IO_DIO5.MAP=5D // map pin DIO5 to IR1
```

The event counter can be enabled or disabled with `MD_ULP_ALG_CTRL.CNT_EN` or `MD_RD_ALG_CTRL.CNT_EN`. A disabled event counter will not generate any interrupts. The event counter requires enabled target extraction.

7.1.4 Data access (RX_BUFFER)

The processing of the moving target indication (MTI), applying windowing, the range FFT, and the Doppler FFT is performed inside the RX_BUFFER memory. All other processing steps, described in Section Algorithm selection, do not modify the content of the RX_BUFFER memory. Therefore, the Range Doppler Image (RDI) remains in memory after the processing is finished and can be read out after presence detection (PD) completion for further off-chip post-processing.

The time it takes to finish on-chip processing after acquiring all chirps can be estimated for the scenarios described in Section ROM configuration with: Config 0 ULP: 0.1 ms, Config 0 RD: 3 ms, and Config 1 RD: 2.2 ms. If you require access to the RD map, it can be read out by software via SPI by setting the bitfield `MD_RD_CTRL.RX_BUF_ACC_EN`.

After the RDI is read, you can power down the RX_BUFFER memory to save power. This is typically done by a power down request via `AVG_CTRL.PDRQ`. However, this power down request is gated by power isolation. This can be worked around by following the given sequence:

1. Powering up the digital power domain containing the radar signal processing via `PDC_CTRL.PM=1 (ON)` and wait for 3 µs
2. Powering down the RX_BUFFER SRAM's via `AVG_CTRL.PDRQ=1`
3. Powering down the digital power domain containing the radar signal processing via `PDC_CTRL.PM=2 (OFF)`
4. Setting power domain controller back to automatic mode via `PDC_CTRL.PM=0 (AUTO)`

The on-chip post-processing chain is triggered each time the selected sequence is completed. Algorithms averaging, moving target indication (MTI), applying windowing and range FFT are processed between chirps. Algorithms Doppler FFT, constant false alarm rate (CFAR), merge, local maximum and target extraction are triggered once after the sequence is completed. Ensure that the sequence produces correct radar data. Once the processing has started, there is no option to stop the execution at a specific step. It is possible to copy data from specific processing steps to FIFO for later analysis described in detail in Section Data forwarding (FIFO).

It is recommended to read out the RDI from the RX_BUFFER memory when the on-chip post-processing is finished. To notify the host via an interrupt, a signal is generated after respective range Doppler sequence is over. As the post-processing time is deterministic the host can read the data from the required RX_BUFFER memory after the wait (post-processing) time. The post-processing time is application specific and depends on the activated post-processing steps.

Note: For evaluation of the post-processing time, the signals "algorithm execution start" (IO_CTRL.DMUX0/1_SEL = 27 (MD_ALG_EXEC)) and "algorithm done" (IO_CTRL.DMUX0/1_SEL = 26 (MD_ALG_DONE)) can be used.

To configure the sequence done signal, one of the below listed options can be used:

- **Using the interrupt router and enabling the range Doppler sequence (IR0/1_EN.EN8/9/10/11):** For example, if interrupt router 0 and sequence 2 is used for the range Doppler, then IR0_EN.EN[10] = 1. This configuration permanently sets the sequence done interrupt and the software must clear the interrupt status via IR0/1_CLR
- **Using the digital multiplexer 2 or 3 and selecting the sequence done pulse (IO_CTLR.DMUX2/3_SEL=28/29/30/31):** With this option, a single pulse is generated when the respective sequence is finished. In such cases, the signal must not be cleared. If a pulse with multiple system clock cycles is required due to timing constraints on the host side, there is an option to extend the pulse via (IO_DIO#.PULSE_LEN)

Each receiver (RX) has a separate RX_BUFFER memory with a specific address range accessible via SPI and is depicted in Table 26. The relative address of the data of a cell is calculated by adding the address shift of its row and its column in the range Doppler image. Finally, the start address offset of the memory for the respective RX antenna needs to be added.

Each cell contains a complex number in cartesian coordinates, with the real part located at address 0x0 and the imaginary part at address 0x4. The data is 24-bit wide and additional zeros are appended to have 32-bit SPI transfer.

Table 26 RX_BUFFER memory address range

RX Buffer	Start address	End address
RX1_BUFFER	0x8000	0x9FFF
RX2_BUFFER	0xA000	0xBFFF
RX3_BUFFER	0xC000	0xDFFF

The organization of the data in the SRAM is shown in Figure 89.

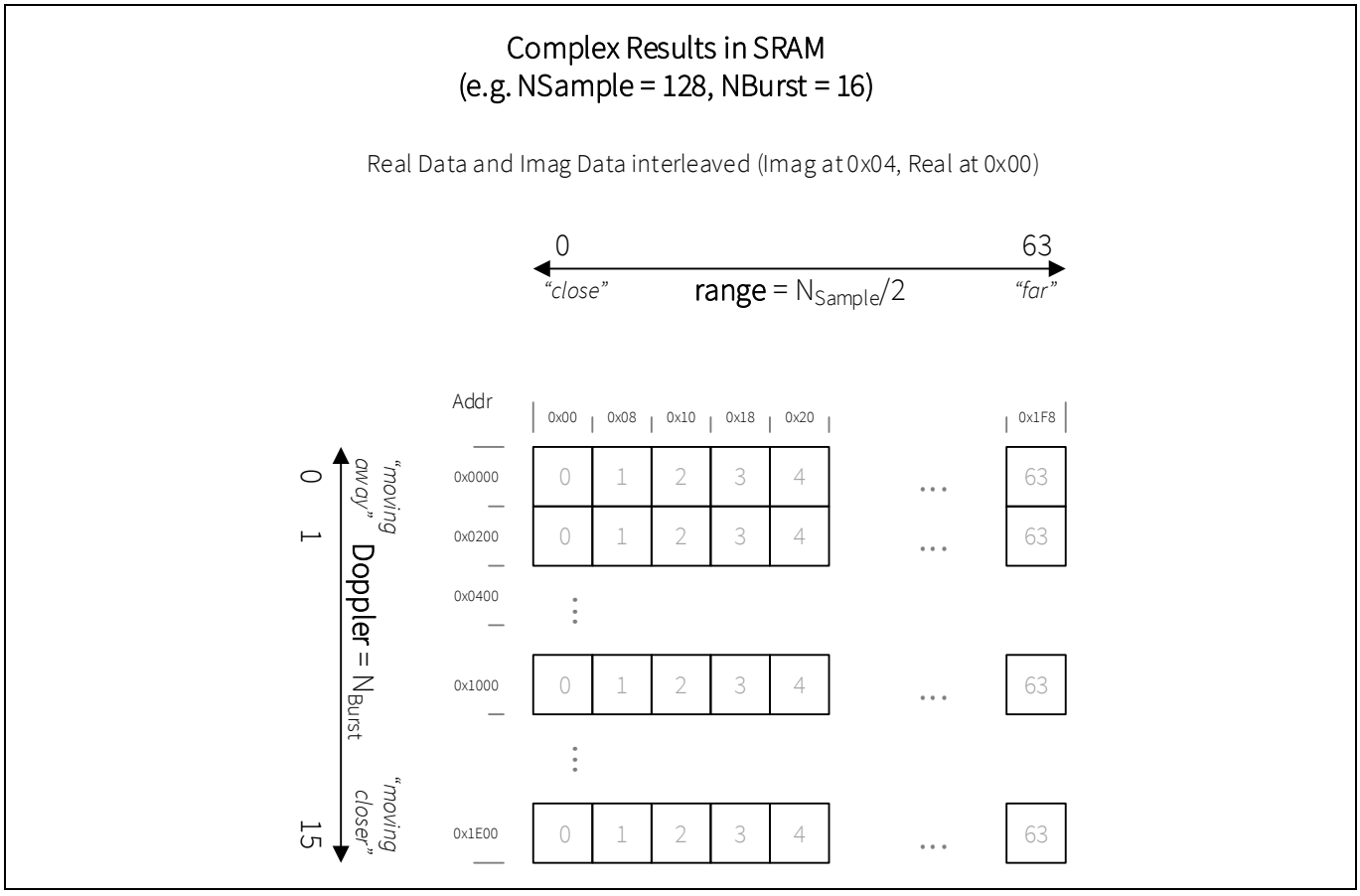


Figure 89 RX_BUFFER data organization

7.1.5 Data forwarding (FIFO)

The device provides the option to copy the results of some processing step from the RX_BUFFER to the FIFO. This enables the host application to gather processing data (such as RFFT raw data, CFAR results) over several frames for later off-chip post-processing or statistics.

Intermediate data dumps can be enabled by setting various bitfields of registers `MD_ULP_FIFO_CTRL` or `MD_RD_FIFO_CTRL`. If a data dump is enabled, the corresponding algorithm (`MD_ULP_ALG_CTRL` or `MD_RD_ALG_CTRL`) needs to be enabled as well.

Note: If the data in the FIFO is only relevant for cases with detected motion, there is an option to clear the FIFO after the motion detection processing is completed without a detected motion (`MD_ULP_FIFO_CTRL.FIFOCLR_EN` or `MD_RD_FIFO_CTRL.FIFOCLR_EN`). All FIFO data will be erased if enabled and no motion was detected. It is recommended to not use this feature together with the FIFO threshold level interrupt feature (FIFO_THRESHOLD).

Note: Forwarding of ADC average data (`MD_ULP_FIFO_CTRL.AVG_EN` or `MD_RD_FIFO_CTRL.AVG_EN`) with "enabled" MTI is not possible.

7.1.5.1 Timing

Data forwarding is triggered after the completion of the corresponding processing step. Therefore, the processing time is increased based on the number of stages enabled which need to be forwarded to FIFO and the number of processed RX channels. The data copy will be done per RX_BUFFER and will copy the data as it is. Starting with the lowest number of processed RX channels, followed by the next higher one. For example, this includes all RX1 samples, all RX2 samples, all RX3 samples, assuming all three RX channels are enabled. In case only RX2 and RX3 are enabled, all RX2 samples are copied first and all RX3 samples are copied afterwards.

Note: If data from processing steps such as AVG, MTI, windowing of RFFT is forwarded (copied from RX_BUFFER) to FIFO, the burst-to-burst time must be long enough. Otherwise, the data (burst) processing will be corrupted. This timing must be ensured by the user software.

The DMA copy duration can be estimated with 1.5 times the number of datapoints plus some additional overhead: $\#cycles \approx 1.5 \times \#transactions + 15$. The number of datapoints depends on the algorithm and radar frame configuration. Table 27 lists the number of datapoints per algorithm, with Sample count per burst NSample, Burst count as NBurst (2^x with disabled MTI or $2^x + 1$ with enabled MTI), number of enabled RX channel NRX. Example of FIFO layout with RFFT and CFFT data are shown in Figure 95 to Figure 98.

Table 27 DMA transaction count for algorithms in units of 32-bit words

Algorithm	Transaction count
AVG	$NSample \times NBurst \times NRX$
MTI	$NSample \times (NBurst - MTI_{EN}) \times NRX$
Windowing	$NSample \times (NBurst - MTI_{EN}) \times NRX$
RFFT	$NSample \times (NBurst - MTI_{EN}) \times NRX$
CFFT	$NSample \times (NBurst - MTI_{EN}) \times NRX$
CFAR	$\frac{NSample \times (NBurst - MTI_{EN})}{32} \times NRX$
RX Merge	$\frac{NSample \times (NBurst - MTI_{EN})}{32}$

Note: Data forwarding described in this chapter applies only with presence detection enabled. If presence detection is disabled, see Section First-in first-out (FIFO) – Data memory.

The runtime for algorithms windowing until RFFT, which are executed between averaged chirps is listed in Table 28. They can be used for ensuring burst-to-burst timing requirements. All channels are processed in parallel.

Table 28 Cycle count runtime for AVG, MTI, windowing and RFFT per burst

Algorithm	Cycle count
AVG	0 - parallel to sampling
MTI	$NSample \times 4 + 10$
Windowing	$\frac{NSample}{4} \times 5 + 10$
RFFT	$NSample \times \log_2(NSample) + NSample + 10$

7.1.5.2 Data layout

This section describes how the data will be written into FIFO when the data forwarding is enabled.

Data is copied to the FIFO immediately after the respective algorithm has been completed. Data dumps will be stored without any additional headers. The software must ensure the data is interpreted correctly in case of multiple processing stages or mixed raw data with motion detect data with interleaved sequences.

To minimize presence detection latency and reduce active time, the processing is interleaved with the data acquisition. The flowchart of the processing steps is shown in Figure 64 and Figure 69. The trigger of the processing steps and its data dumps are listed in Table 29.

Table 29 Processing/data dump timing

Algorithm	Processing Time
AVG	with chirp acquisition
MTI	after AVG, starting with 2nd burst
Windowing	after MTI, starting with 2nd burst
RFFT	after Windowing, starting with 3rd burst
CFFT, CFAR, MERGE	after frame completion

An example of a data dump with 17 bursts, a single RX and an enabled data forwarding of AVG, MTI, RFFT, and CFFT is shown in Figure 90.

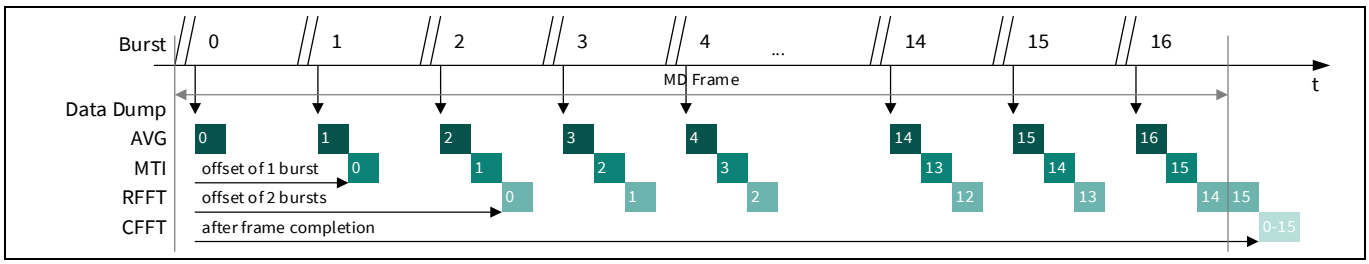


Figure 90 Processing/data dump timing

The AVG processing and its data dump is triggered with the completion of the first burst. The first MTI is triggered with the completion of the second burst. The RFFT is triggered with the completion of the third burst, whereas the remaining last two RFFTs are triggered together. The CFFT and all subsequent processing steps are triggered with an MD frame completion.

An overview of the data layout inside FIFO is depicted in Figure 91 and Figure 92. It shows a hypothetical RD sequence with 5 bursts (4 after MTI) with all algorithms enabled. Disabled antennas or algorithms will not write data to the FIFO. Algorithms AVG until CFAR have one result per active RX channel, merge has one result in total for all aggregated channels. The first windowing and RFFT execution is deferred by one burst with respect to the MTI. AVG data cannot be forwarded if MTI is enabled, but still both algorithms are depicted in the following diagrams for presentation purposes. Nomenclature according to Figure 69.

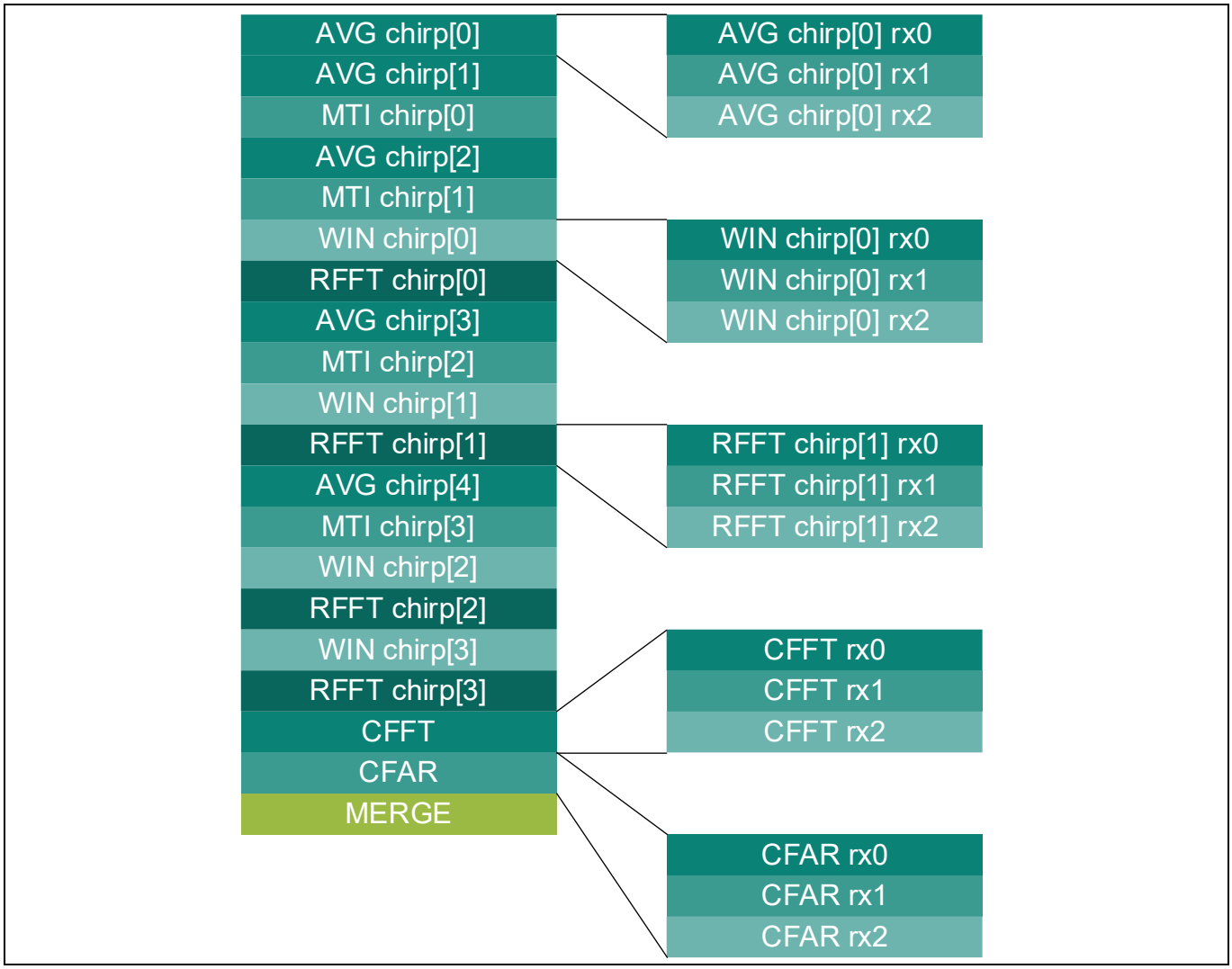


Figure 91 Data ordering inside FIFO for DMA data with enabled MTI



Figure 92 Data ordering inside FIFO for DMA data with disabled MTI

The width of the forwarded data is always 32-bit. The 24-bit processed data in the RX_BUFFER is extended by a zero padded byte on LSB or MSB side, selected by `SPI_CTRL.MSB_ALIGN_EN`. Binary target data after CFAR is only 16-bit wide and padded with an additional byte on the MSB side to form the 24-bit postprocessing word.

The layout inside a region depends on the algorithm. Samples of AVG, MTI and WIN are in bit-reverse ordering. They are pairwise grouped, with one even and one odd sample per group. Bit-reversed order is created by reversing the sample pair number bit sequence, such as averaged chirp with 16 samples --> 3 bits are needed for numbering the eight sample pairs. 0th sample pair: rev(0b000) = 0b000; 1st sample pair: rev(0b001) = 0b100; 2nd sample pair: rev(0b010) = 0b010; 3rd sample pair: rev(0b011) = 0b110; 4th sample pair: rev(0b100) = 0b001; and so on, see Table 30. From RFFT algorithm onwards, the samples are in natural order.

Table 30 Bit-reverse ordering example with N = 128_{10}

Natural		Bit-reverse	
0_{10}	$000\ 0000_2$	$000\ 0000_2$	0_{10}
1_{10}	$000\ 0001_2$	$100\ 0000_2$	64_{10}
2_{10}	$000\ 0010_2$	$010\ 0000_2$	32_{10}
3_{10}	$000\ 0011_2$	$110\ 0000_2$	96_{10}
4_{10}	$000\ 0100_2$	$001\ 0000_2$	16_{10}
…	–	–	…
126_{10}	$111\ 1110_2$	$011\ 1111_2$	63_{10}
127_{10}	$111\ 1111_2$	$111\ 1111_2$	127_{10}

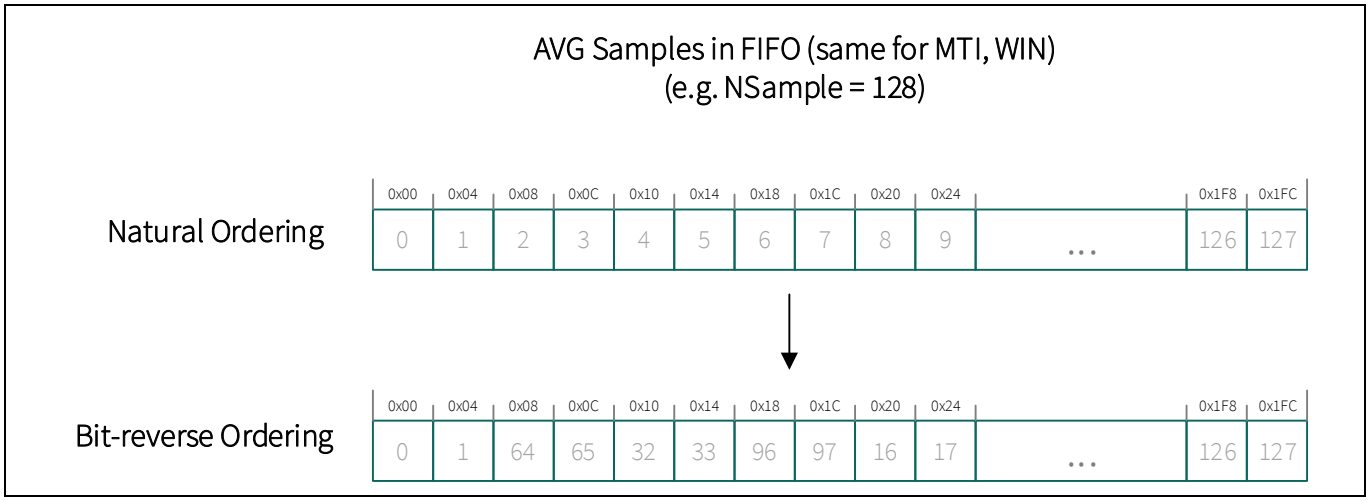


Figure 93 Bit-reverse ordering of samples inside a burst (such as 128 samples)

Averaged data samples are pair-wise bit-reversed with all enabled RX channels grouped together for each averaged chirp, as shown in Figure 93 and Figure 94.



Figure 94 FIFO layout with AVG data for full RD frame (such as 128 samples)

RFFT samples are in natural order with all enabled RX channels for each averaged chirp grouped together. Real and imaginary part of each sample is kept together as well, see Figure 95 and Figure 96.

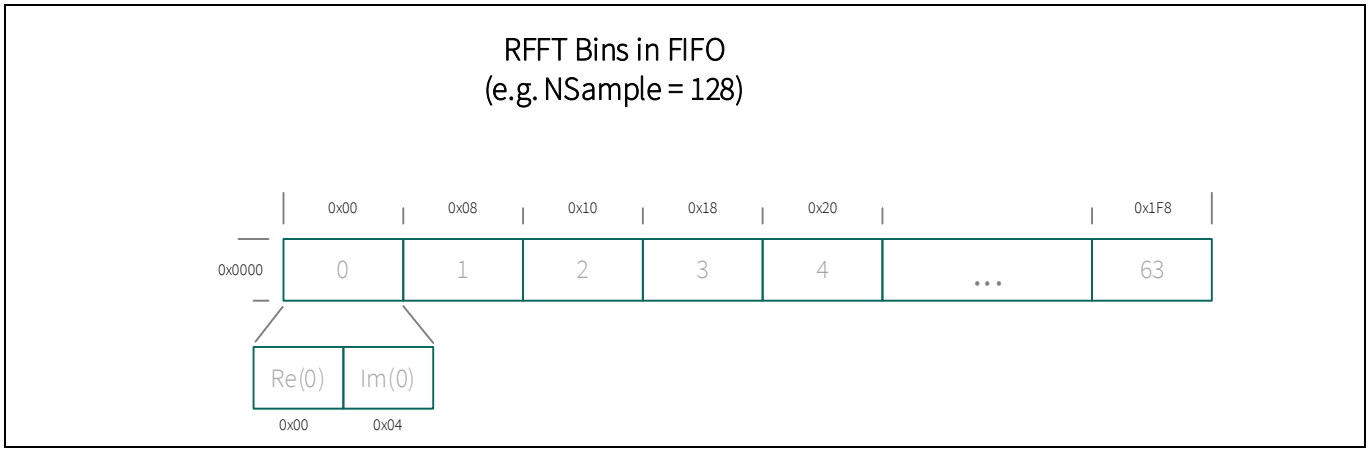


Figure 95 FIFO layout with RFFT data for single averaged chirp

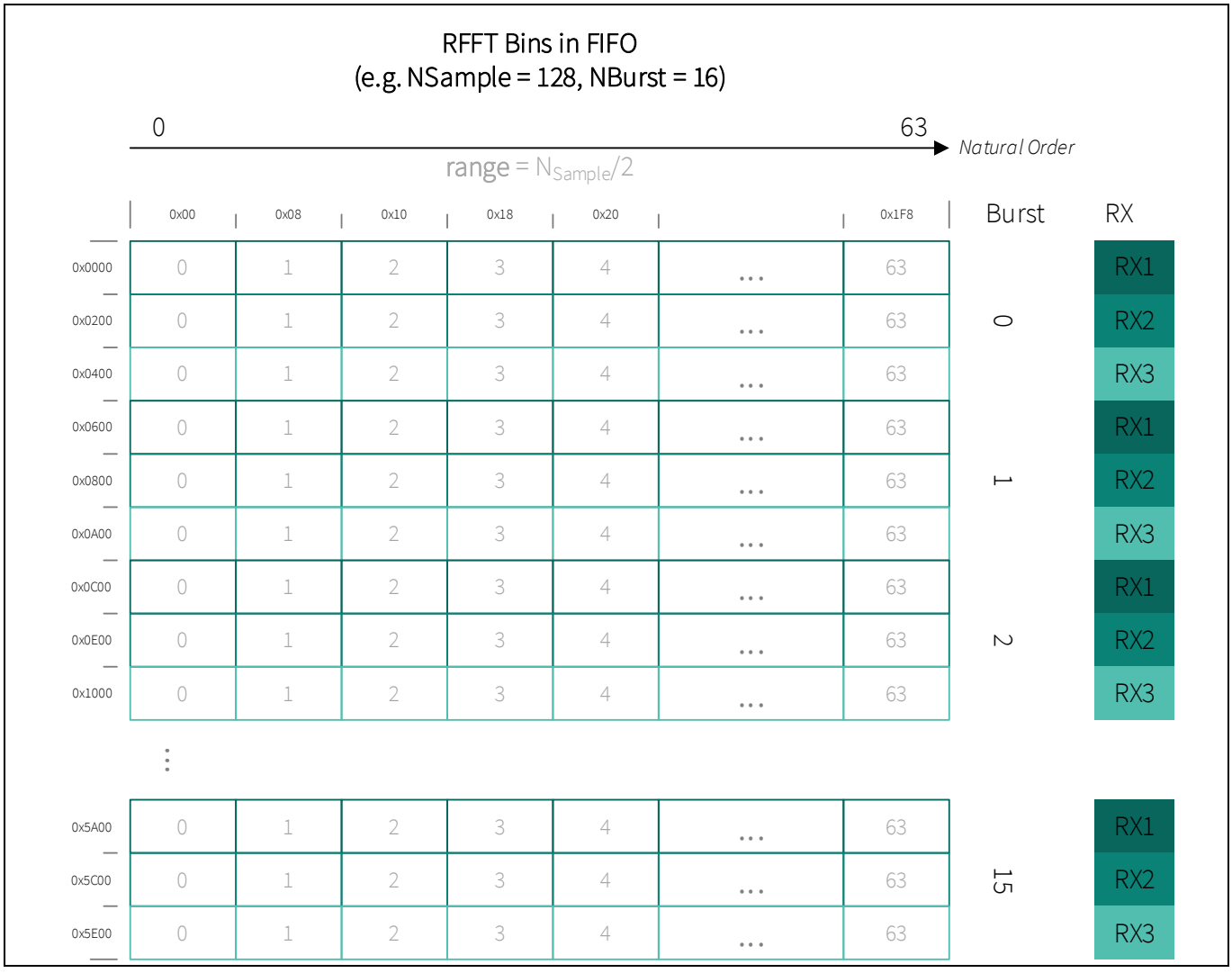


Figure 96 FIFO layout with RFFT data for full RD frame

CFFT data is in normal linear ordering with each enabled RX channel kept together. The first dimension is range. Second dimension is Doppler, as shown in Figure 97.

Note: The CFFT data is forwarded without reordering, meaning the zero frequency is at the edges. To move the zero frequency to the middle of the RD map, a FFT-shift operation has to be applied after reading out the data by swapping the lower and upper halves. Succeeding algorithms like CFAR, merge and local maximum have the zero frequency bins already in the center.

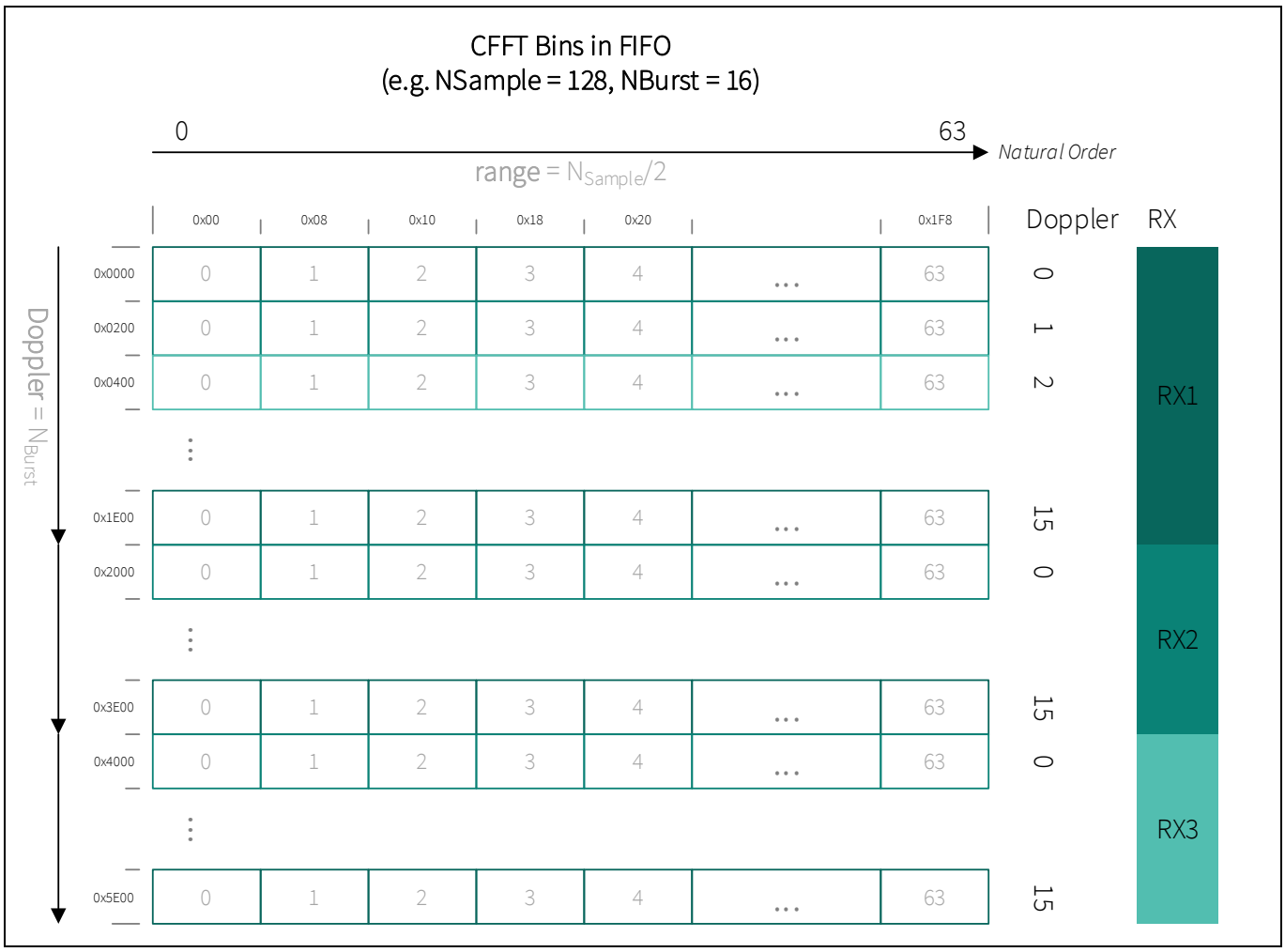


Figure 97 FIFO layout with CFFT data

CFAR and merge data contain binary data only, where one bit corresponds to one bin. Each FIFO word contains 16 binary values, the remaining space is filled with 0, as shown in Figure 98.

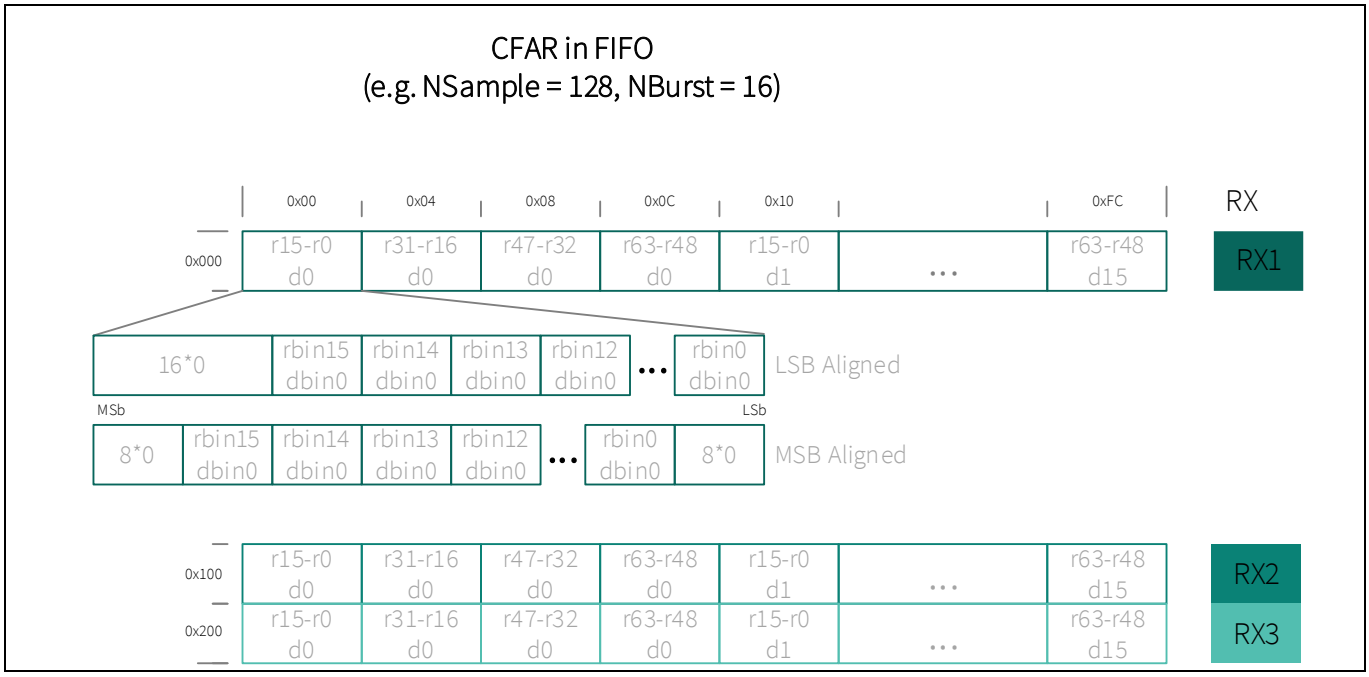


Figure 98 FIFO layout with CFAR data

7.1.6 Clutter handling

The on-chip signal processing chains, ULP and RD, offer various clutter rejection methods to minimize false positives, as shown in the Table 31. By leveraging these features, the device can achieve the lowest power consumption while maintaining optimal performance.

Table 31 On-chip features for clutter rejection

Features	ULP mode	RD Mode	Applications
Range Masking	Yes	Yes	To exclude clutter range bins from target extraction step
Doppler Masking	No	Yes	To exclude slow moving clutters like tree movements, cleaning robots and curtain movements
Event Counter	Yes	Yes	To reduce false positives coming from the random movements in the FoV of radar or strong winds
Angle Masking E & H plane	Yes	Yes	To filter out the targets of certain height
Range Angle zone masking	Yes	Yes	To filter out the target detections from a certain Range Angle zone area (polygons)

8 ROM configuration

8.1 Config 0

This ROM configuration highlights the sensor's ability to address a range of indoor presence detection use cases. By integrating ultra-low power (ULP) and Range Doppler (RD) modes of operation, the sensor effectively detects both static and moving individuals within its field of view. This dual-mode operation ensures reliable and efficient performance.

Designed specifically for indoor applications requiring robust presence detection, this configuration offers a power-saving solution. It operates at an update rate of 4 Hz, corresponding to a frame interval of 250 ms. Figure 99 shows the frame executed with this configuration which includes both ULP and RD sequences, thereby enabling comprehensive coverage in diverse presence detection scenarios.



Figure 99 Frame Structure containing ultra-low power and range Doppler sequence

Figure 100 shows the arrangement of RD and ULP bursts within the defined sequences, arranged based on the averaged chirps corresponding to each mode. Once the ADC (analog-to-digital converter) calibration is completed, the GTMR0 timer is activated. The GTMR0 timer gets triggered periodically with an interval of 250 ms. Within this 250 ms interval, the GTMR0 timer sequentially initiates both RD and ULP sequences. Specifically, when GTMR0 gets triggered, there is a 2 ms delay before the RD sequence begins execution.
This 2 ms delay provides a possibility to reconfigure some registers via SPI write from external host if necessary. Additionally, the ULP sequence is triggered twice within this interval, occurring at every 100 ms. This timing configuration ensures efficient and systematic execution of both RD and ULP modes, facilitating optimal detection across varying application scenarios.

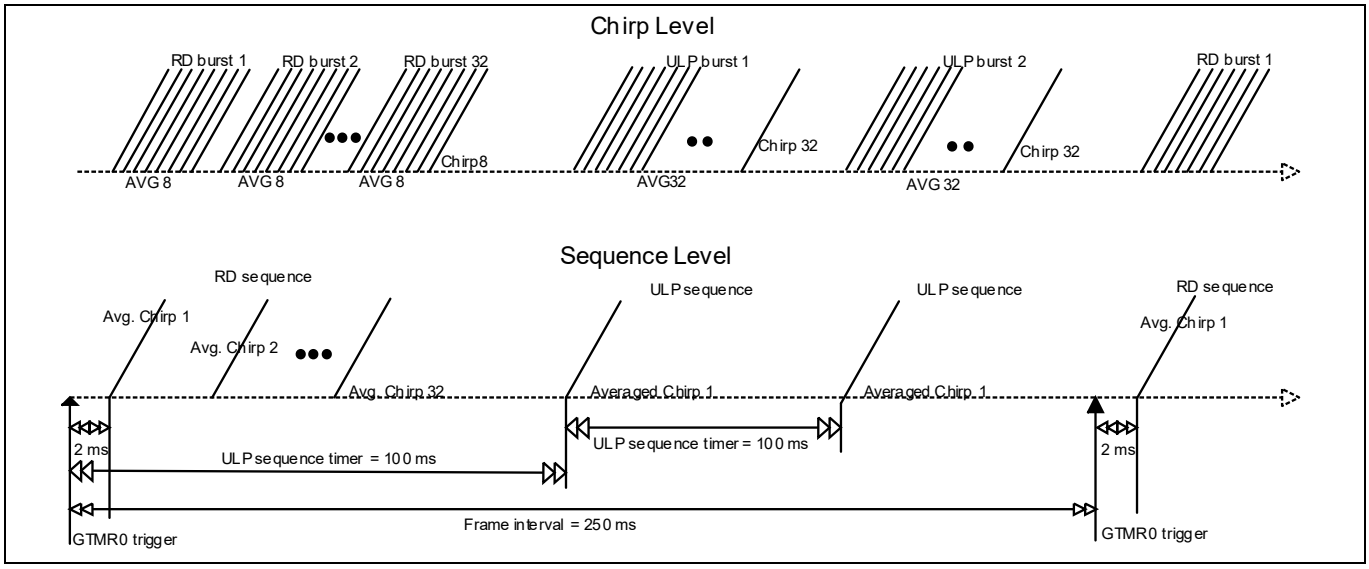


Figure 100 Timing model

Table 32 Doppler signatures

Sequence type	Minimum Doppler speed (m/s)	Maximum Doppler speed (m/s)
Range-Doppler (RD)	–2.38 < x < –0.14	0.14 < x < 2.35
Ultra-low power (ULP)	–0.0122	0.01

The results presented in Table 32 demonstrate the minimum and maximum Doppler speeds achievable with the RD and ULP sequences in this configuration. To mitigate the effects of slow-moving clutter, Doppler masking is applied to Doppler bins adjacent to the zero-Doppler bin, effectively reducing the likelihood of false positives and enhancing the overall accuracy of the system.

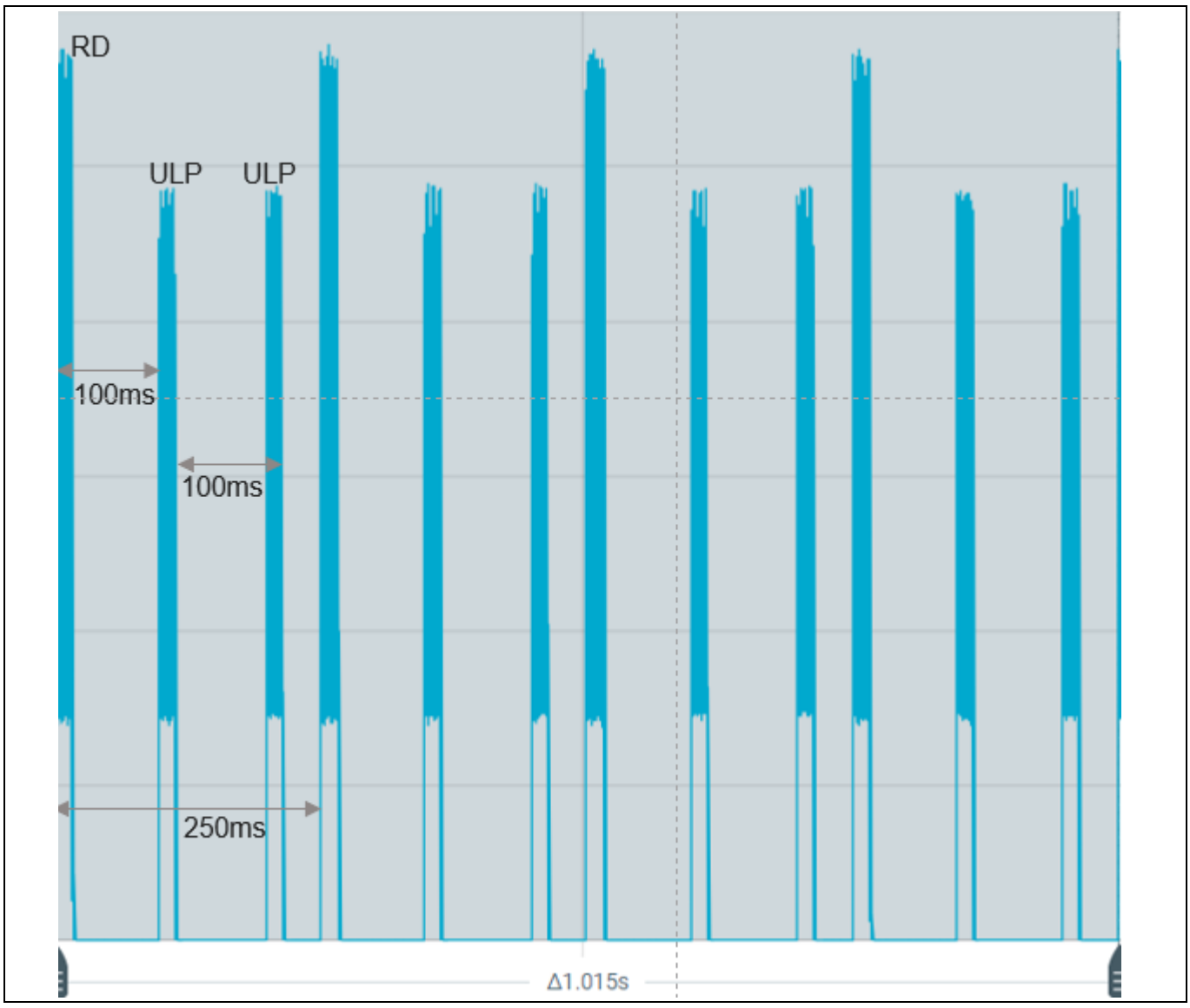


Figure 101 Nordic Power Profiler Kit showing the modulation scheme for a period of 1 second

The modulation scheme used in this configuration is shown in Figure 101, which is observed using the Nordic Power Profiler Kit (NRF-PPK2). The calculated power values for this setup are summarized in Table 33. However, it is important to acknowledge that the Nordic Power Profiler Kit is subject to a constraint that prevents it from accurately capturing the maximum peak of the modulation. Therefore, the power values presented here should be considered as approximate estimates.

Note: *NRF-PPK2 is a registered trademark of Nordic Semiconductor. The use of NRF-PPK2 results in this document does not imply any endorsement or affiliation with Nordic Semiconductor. All intellectual property rights related to the NRF-PPK2 tool, and its underlying technology are owned by Nordic Semiconductor.*

Table 33 Power values

Voltage (V)	Current consumption (mA)	Power (mW)
1.2	19.4	23.3
1.8	2.7	4.9
Total	–	28.2

8.1.1 Field of view experiment for maximum detectable range validation

The provided configuration has been tested to determine the sensor's maximum detectable range, with results validated through field of view (FoV) experiments. The objective of this experiment is to evaluate range Doppler (RD) detections across the entire field of view.

Figure 102 shows the experiment setup, where the sensor is placed in its E-plane configuration and the target performs a broadside walk, moving towards and away from the sensor. This procedure is conducted for angular positions ranging from –70° to +70° relative to the sensor, in 10° increments.

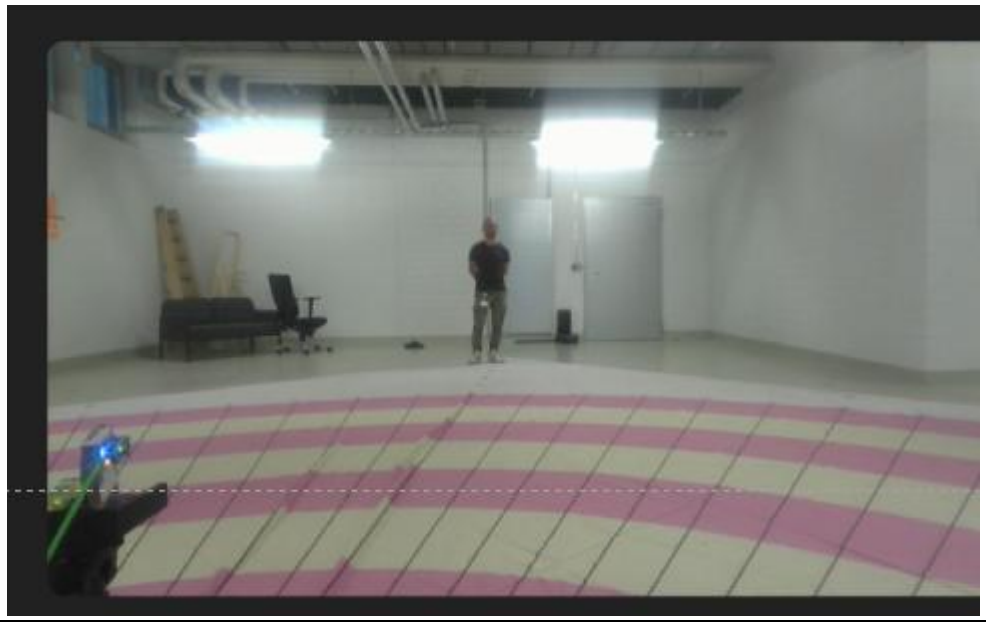


Figure 102 Experiment setup

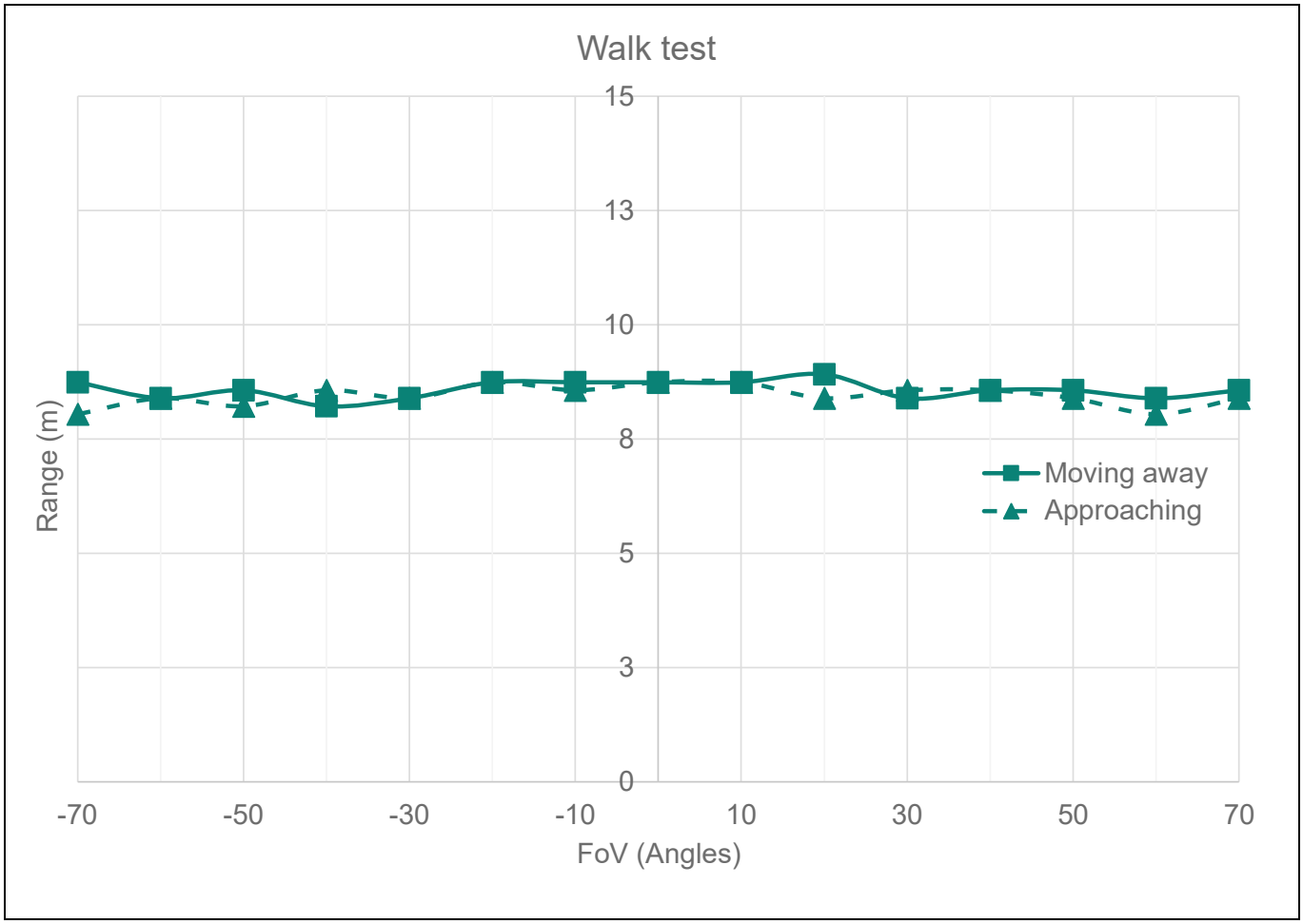


Figure 103 RD target maximum range detections over the entire field of view (FoV)

Figure 103 shows the RD maximum range detections when the target moves away and approaches the sensor. Continuous RD range detections are observed until a certain range from the sensor, based on the constant false alarm rate (CFAR) parameters set in this configuration. These results demonstrate that the targets can be reliably detected up to a maximum range of 9 meters across all angular sectors with this configuration.

This experiment confirms the robust detection capabilities of the sensor within the specified maximum range and across the defined field of view angle sectors.

8.1.2 Operating modes

This configuration supports several operating modes that can be used with a host microcontroller (MCU), relay circuitry, or via SPI commands from host MCU. The available modes are described below:

8.1.2.1 Hardware reset/power cycle (no SPI required)

The interrupt generated by the radar can be cleared by the host microcontroller or relay in one of the following ways:

- Performing a hardware reset, or
- Powering OFF the radar upon receiving the interrupt, then powering it back ON after a specific time interval as shown in Figure 104

No SPI communication from host MCU is required during this mode of operation.

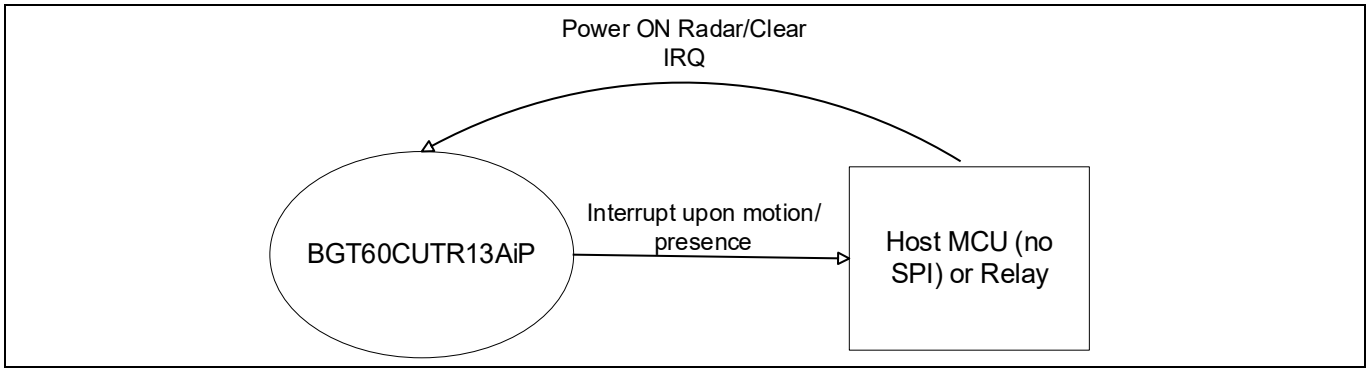


Figure 104 IRQ clear with Hardware reset

8.1.2.2 Interrupt cleared via SPI command

The interrupt generated by the radar is cleared by the host microcontroller by sending an SPI command. No hardware reset or power cycling is needed as illustrated in Figure 105

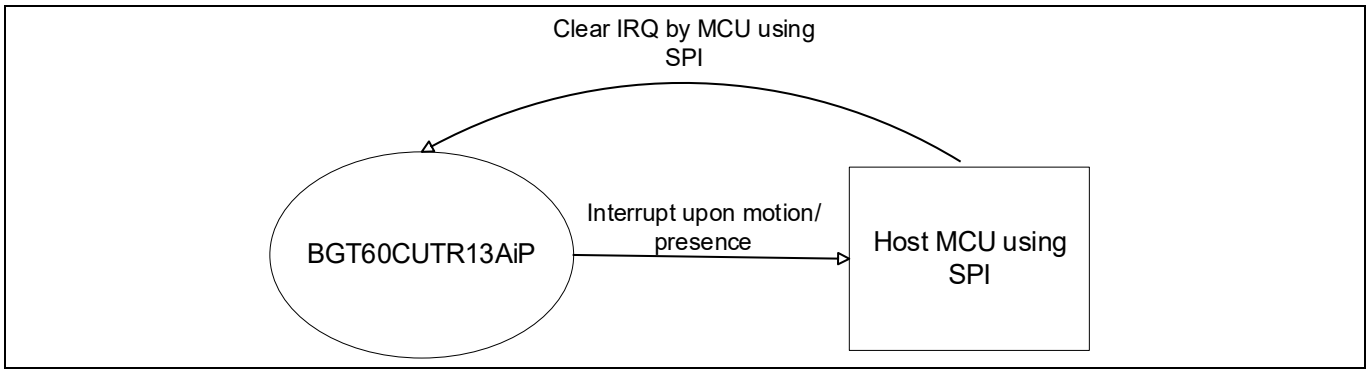


Figure 105 IRQ clear using SPI from host MCU

8.1.2.3 Motion-triggered power control via SPI

When a motion interrupt occurs, the host microcontroller is responsible for:

- Powering ON the radar and triggering the range-Doppler sequence to detect motion within a minimum time interval of 21 ms (SEQ2 active RF time plus MD processing time), or
- Powering OFF the radar if no motion interrupt is observed for the minimum time interval of 21 ms

This mode is controlled using SPI commands as described in Figure 106

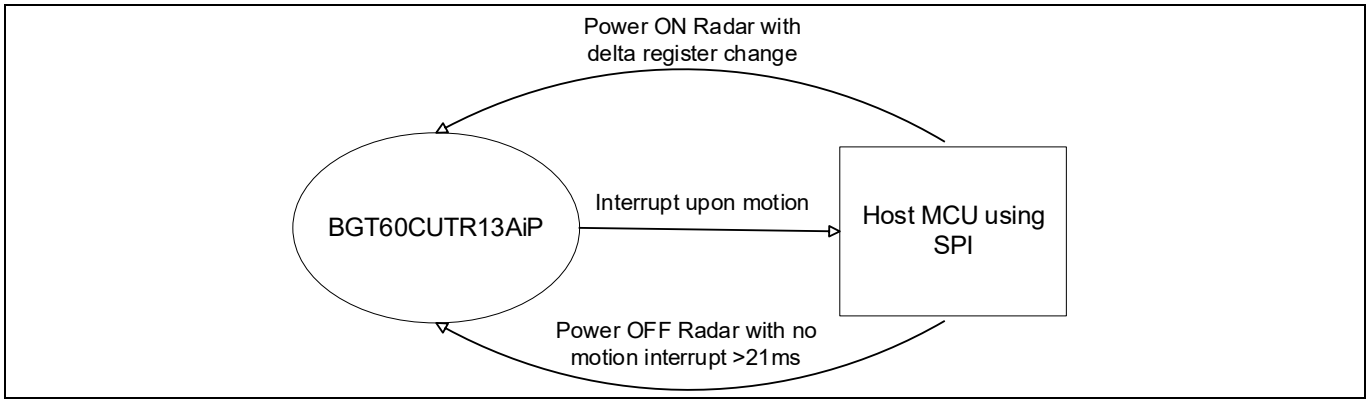


Figure 106 Motion interrupt-based power control using SPI from Host MCU

Note: The delta register changes need to be done to operate ROM config 0 as only motion detector.

Code listing 13 Delta register changes to operate ROM config 0 as only motion detector

```
MD_RD_EVENT   0x1de0 0x01010100   //Event counter taken care by Host MCU
MD_ULP_CTRL   0x1c00 0x07070202   //Disable ULP mode from processing
GTMR0_CTRL    0x0600 0x000b0000   //Disable automatic triggering using RCOSC
GTMR1_CTRL    0x0610 0x00040200   //Disable GTMR1 from triggering ULP
GTMR2_CTRL    0x0620 0x00040100   //Disable GTMR2 from triggering RD
CLK_WUC       0x0030 0x00000800   //Trigger RD mode (SEQ2) using the Host MCU
```

8.1.2.4 Interrupt-based power cycling

When an interrupt is triggered, the microcontroller or relay is responsible for powering the radar ON or OFF for a minimum time interval of 1.3 seconds (ULP Event Counter Threshold of 5 times ULP update rate of 250 ms plus the processing time). This time interval allows the system to detect either a motion or presence interrupt accordingly as shown in Figure 107.

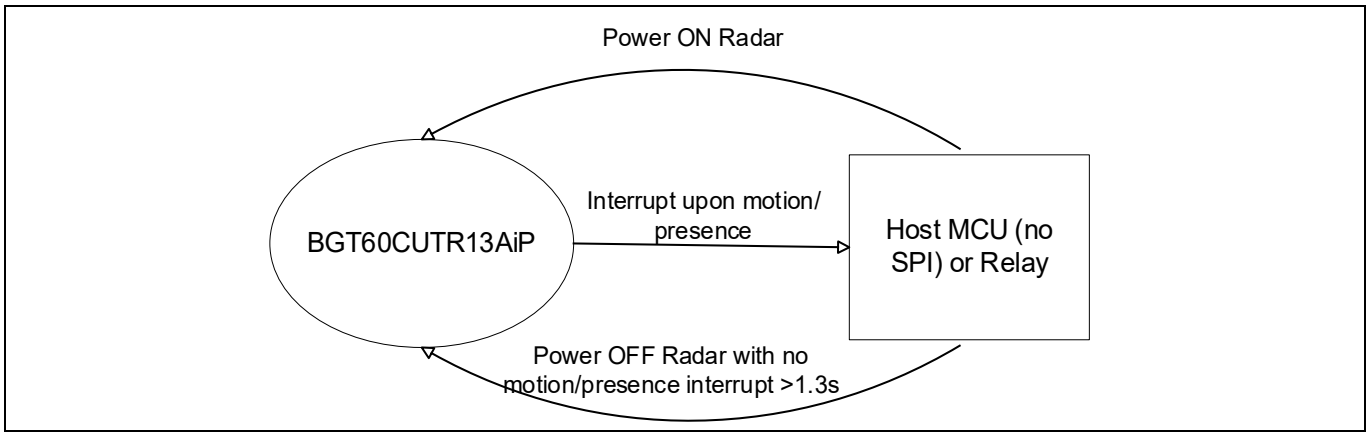


Figure 107 Interrupt based power cycling concept with no SPI involved from host MCU

8.2 Config 1

This configuration utilizes the range Doppler sequence (SEQ2), which is designed to operate at two different update rates to suit varying outdoor application scenarios. The configuration begins with analog-to-digital converter (ADC) calibration, after which the RD sequence (SEQ2) is initiated. Following the completion of SEQ2, the GTMR2 timer is triggered, setting the RD sequence to execute with a period of 1 Hz. When no motion is detected, the RD sequence maintains this 1 Hz update rate, as shown in Figure 108.

However, if motion is detected during the RD sequence, the `MD_RD_EVENT` interrupt is generated. This interrupt activates another timer, GTMR0, which switches the RD sequence to a higher update rate of 10 Hz to capture motion detection more effectively. Both the GTMR0 and GTMR2 timers are equipped with retrigger functionality. This feature ensures that if motion continues to be detected, the `MD_RD_EVENT` interrupt resets the GTMR0 timer, allowing the RD sequence to persist at the high 10 Hz update rate.

When no motion is detected for the last 10 RD sequences, the configuration automatically reverts to the 1 Hz update rate, ensuring efficient power utilization. This dynamic switching mechanism, as shown in Figure 108 enables the configuration to adapt seamlessly between situations with and without motion.

This setup is specifically tailored for outdoor use cases, such as smart security applications. For instance, in a security doorbell system, the sensor operates in low-power 1 Hz RD mode while monitoring its environment. Once a target enters its field of view, the sensor transitions to the high update rate of 10 Hz, generating more frequent RD detections and interrupts. This significantly reduces latency to detect the target, ensuring rapid response and precise motion capture. Figure 109 shows the timing model for this range Doppler configuration.

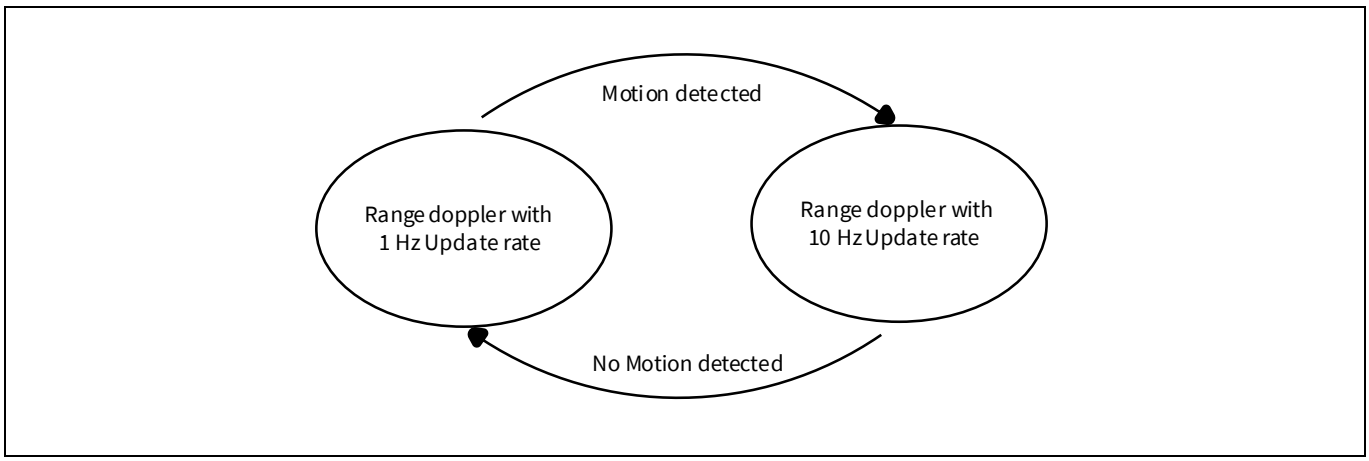


Figure 108 Frame structure with RD sequences operating with two different update rates

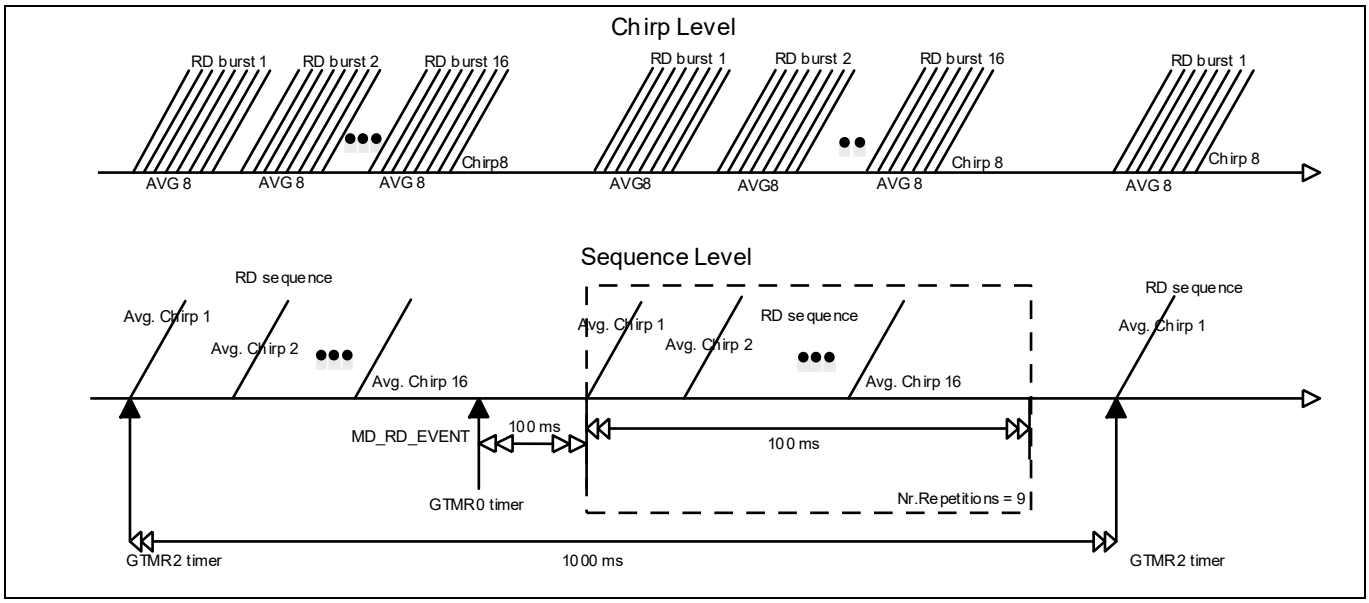


Figure 109 Timing model

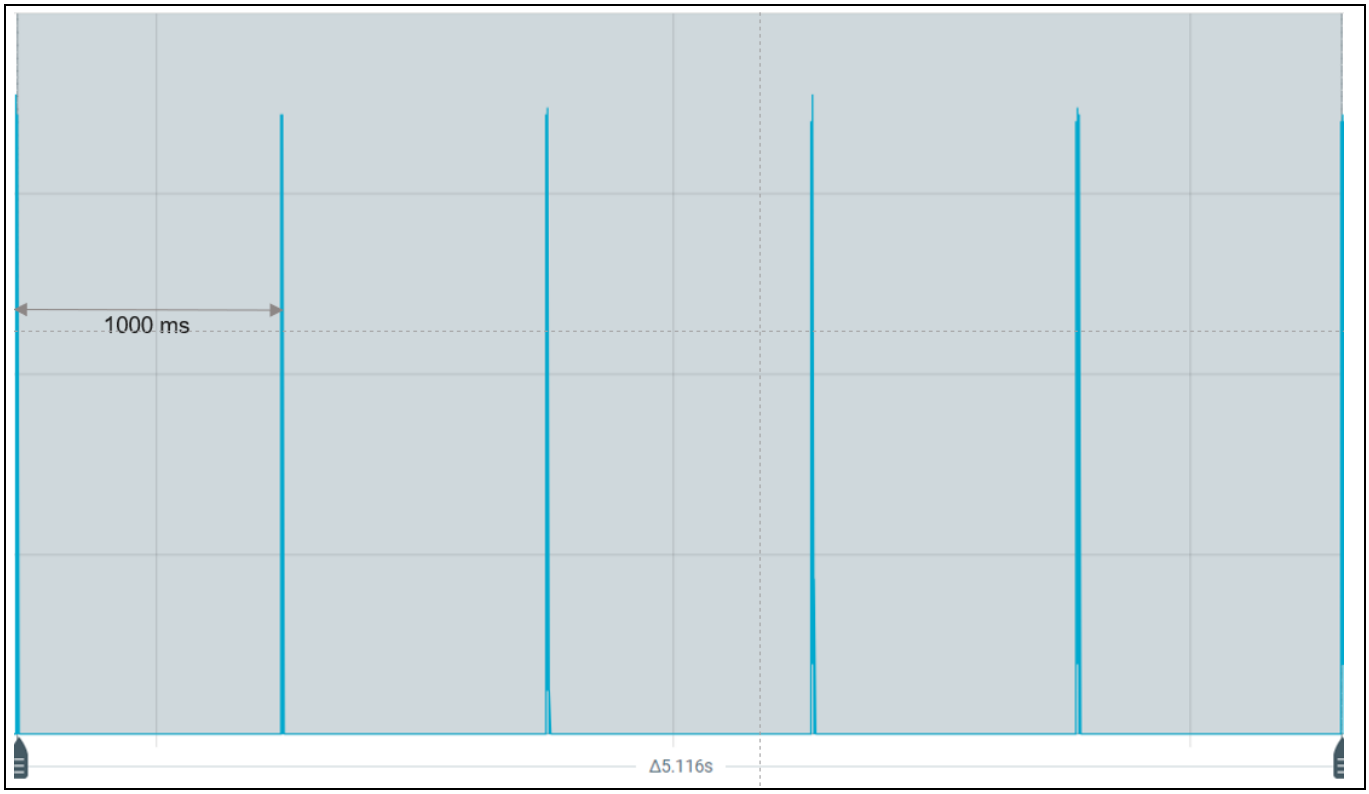


Figure 110 Nordic Power Profiler Kit showing the RD sequences running at low fps of 1Hz

Figure 110 shows the RD sequence operating in standby mode, characterized by a low update rate of 1 Hz. In contrast, Figure 111 demonstrates the dynamic behavior of the RD sequence, where it transitions to a high update rate of 10 Hz in response to motion detection, and subsequently reverts back to the 1 Hz update rate when motion is no longer detected. The Power Profile Kit is used to capture this dynamic behavior, and the resulting power consumption values for the RD sequence in standby mode are presented in Table 34. These values provide a quantitative assessment of the sequence's power efficiency during periods of inactivity.

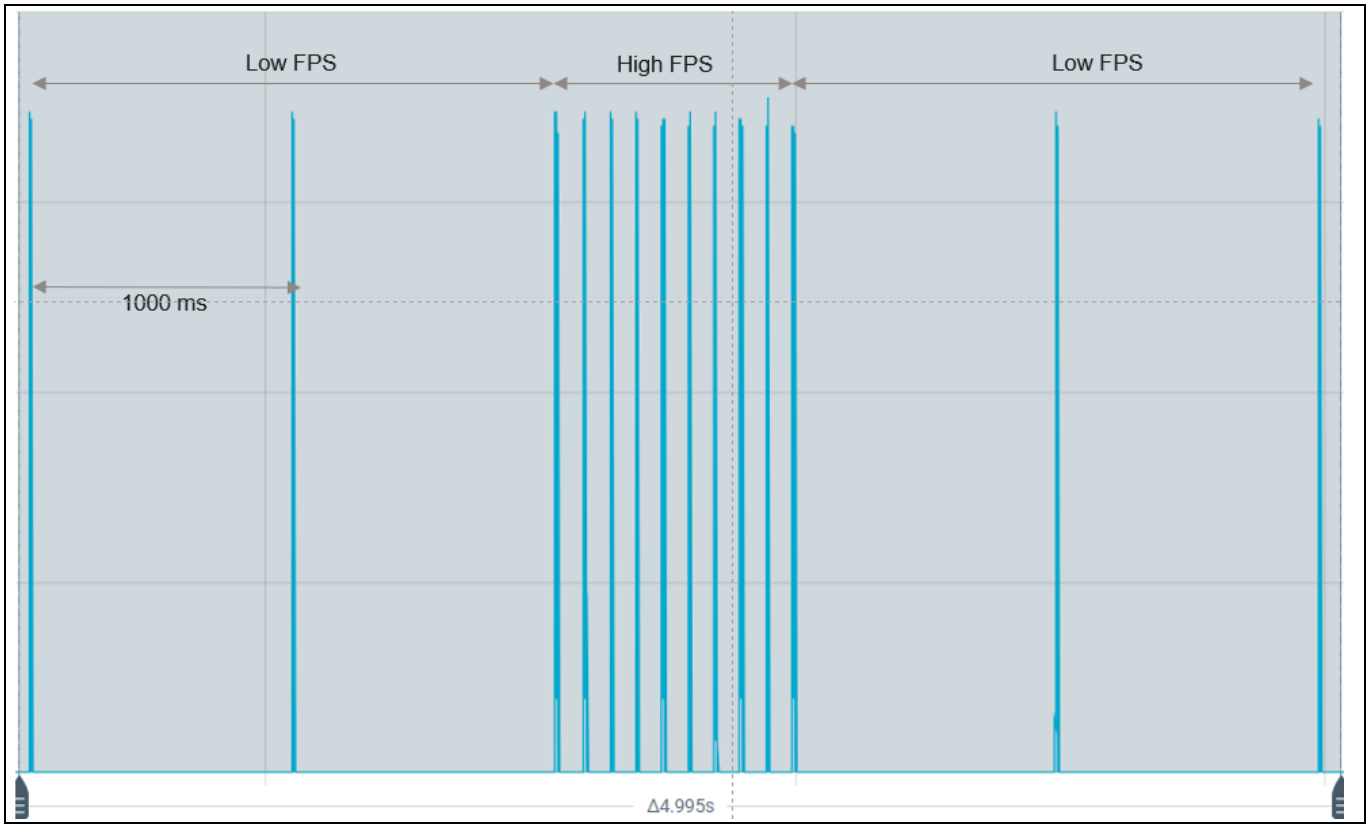


Figure 111 Nordic Power Profiler Kit showing the RD sequences running at low fps and switching to high fps after the MD_RD_EVENT

Table 34 Power values at standby mode (low fps)

Voltage (V)	Current (mA)	Power (mW)
1.2	1.2	1.4
1.8	0.2	0.4
Total	–	1.8

8.2.1 Outdoor motion detection experiment and performance validation

This configuration has been specifically optimized for outdoor motion detection use cases. To validate its performance, the experiment is conducted in an outdoor environment with cluttered surroundings, including factors such as strong winds and moving objects like trees. These conditions mimic real-world scenarios where robust motion detection is required so that it avoids the trigger coming from these false targets (clutters).

In the setup, the sensor is placed in its E-plane configuration while the target performs a broadside walk toward and away from the sensor. This procedure is repeated for angular positions ranging from –70° to +70° relative to the sensor, with measurements taken at 10° increments.

To minimize false positives caused by environmental clutter, this configuration employs a constant false alarm rate (CFAR) threshold along with Doppler masking. Doppler masking eliminates the zero Doppler component (stationary objects) as well as low Doppler components, improving the detection accuracy for moving targets.

Figure 112 shows the observed Range Doppler (RD) targets maximum range detections during these experiments. These results demonstrate the maximum range that can be achieved when the target moves away and approaches the sensor across different angle sectors in the field of view. This experiment confirms the configuration's suitability for outdoor scenarios, even under challenging environmental conditions, ensuring accurate motion detection.

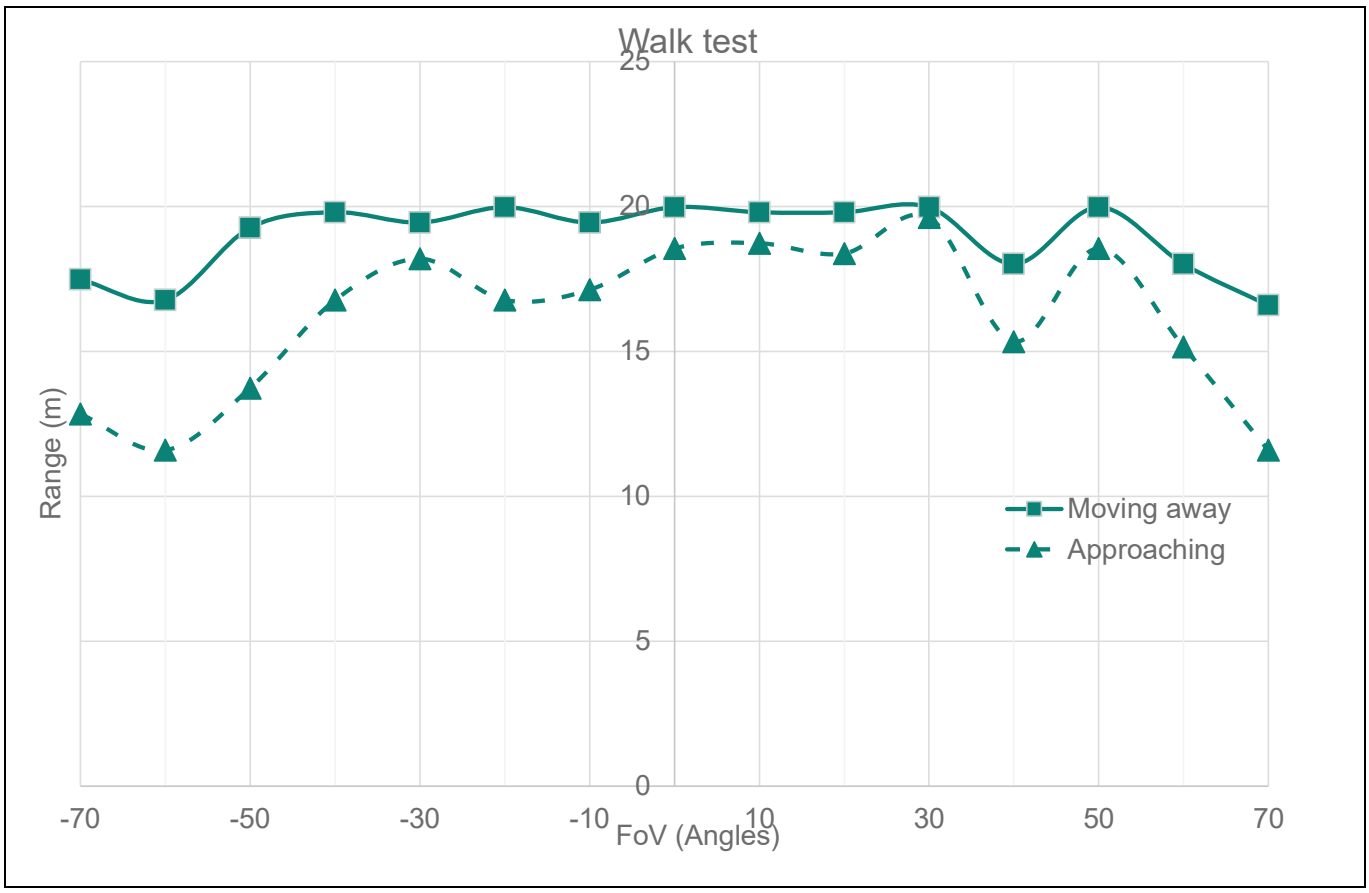


Figure 112 RD target maximum range detections over the entire field of view in outdoor environment

8.2.2 Operating modes

This configuration supports multiple operating modes, giving you the flexibility to work with a host microcontroller, relay circuitry, or sending SPI commands from host MCU based on your application needs. The sections below describe each available mode and how the system responds to radar interrupts.

8.2.2.1 Hardware reset/power cycle (no SPI required)

The interrupt generated by the radar is cleared by the host microcontroller or relay in one of the following ways:

- Performing a hardware reset, or
- Powering OFF the radar upon receiving the interrupt, then powering it back ON after a specific time interval.

No SPI communication from host MCU is required for this mode of operation as shown in Figure 113.

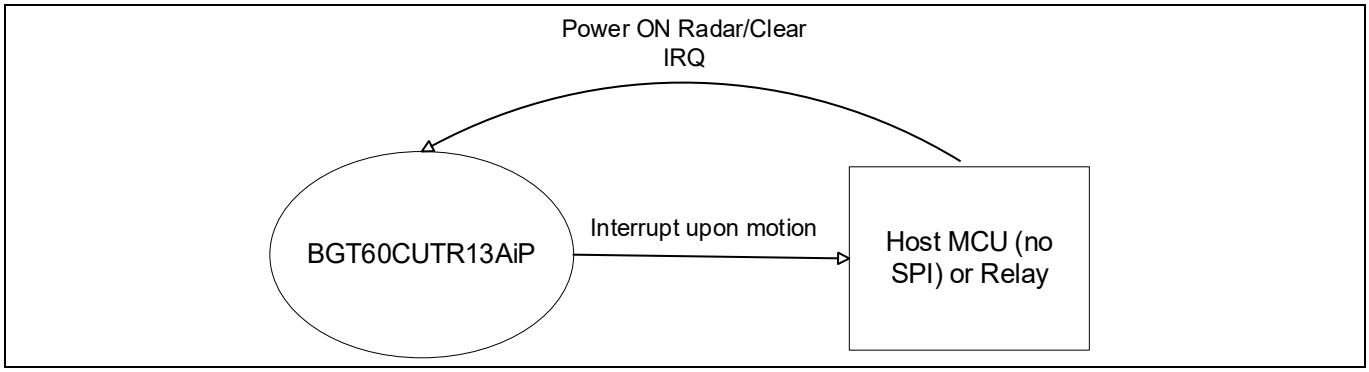


Figure 113 IRQ clear with Hardware reset

8.2.2.2 Motion-triggered power control (no SPI required)

When a motion interrupt is detected, the host microcontroller is responsible for:

- Powering ON the radar and triggering the range-Doppler sequence to confirm motion within a minimum time interval of 15 ms (SEQ2 active RF time plus the MD processing time), or
- Powering OFF the radar if no motion interrupt is observed for the minimum time interval

No SPI communication from host MCU is required for this mode of operation as illustrated in Figure 114

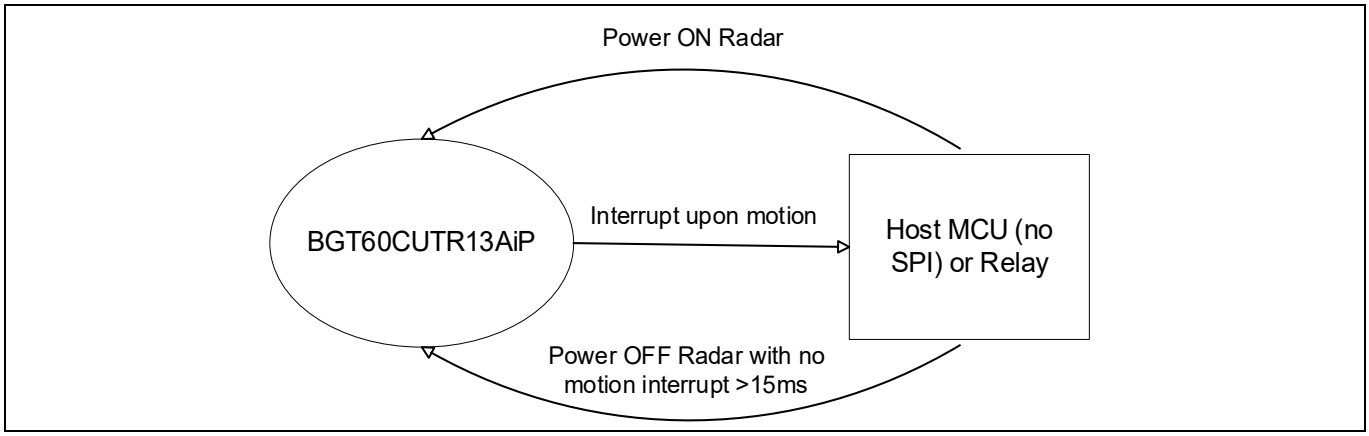


Figure 114 Motion interrupt-based power control with Host MCU

8.2.2.3 Interrupt Handling with High Frame Rate Switching (SPI Required)

When a motion interrupt occurs, the host microcontroller or relay clears the interrupt using an SPI command. Upon receiving the first interrupt, the radar automatically switches from low frame rate (1Hz) to high frame rate (10Hz) mode for further off-chip processing. Figure 115 shows how the host MCU clears the radar interrupt using the SPI command.

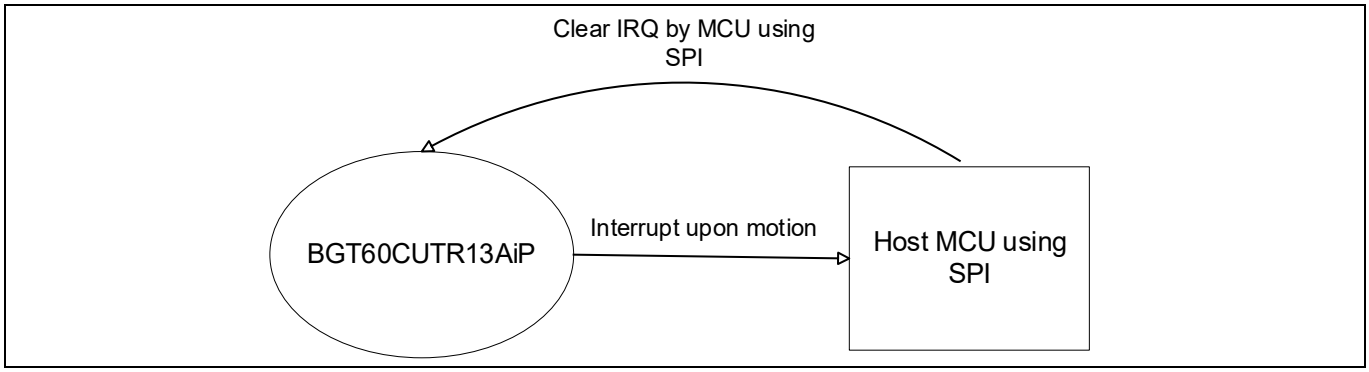


Figure 115 IRQ clear using SPI from host MCU

Note: The high frame rate mode can be disabled at any time by sending an SPI write command from the host microcontroller.

Disable High FPS mode using SPI write GTMR2_CTRL 0x0620 0x0000a008.

Enable High FPS mode using SPI write GTMR2_CTRL 0x0620 0x0000a009.

8.3 Config 2

This configuration operates using the ultra-low power (ULP) mode sequence (SEQ1), which is optimized for energy-efficient motion detection. Once ADC calibration is completed, the GTMR0 timer is activated via a software trigger. The GTMR0 timer runs at an update rate of 2 Hz, triggering two ULP bursts spaced 1 ms apart within this 500 ms time interval as shown in Figure 116.

After the completion of the second ULP burst, the ULP processing is performed. If motion is detected during this processing step, the `MD_ULP_EVENT` interrupt is generated. This interrupt then activates another general timer, GTMR1, which holds the interrupt state for a period of 30 seconds. Figure 117 shows the behavior of this configuration. Also, the GTMR1 timer includes a retrigger functionality, meaning that whenever the `MD_ULP_EVENT` occurs within this 30 second period, the timer is reset and extends the interrupt state to another 30 seconds.

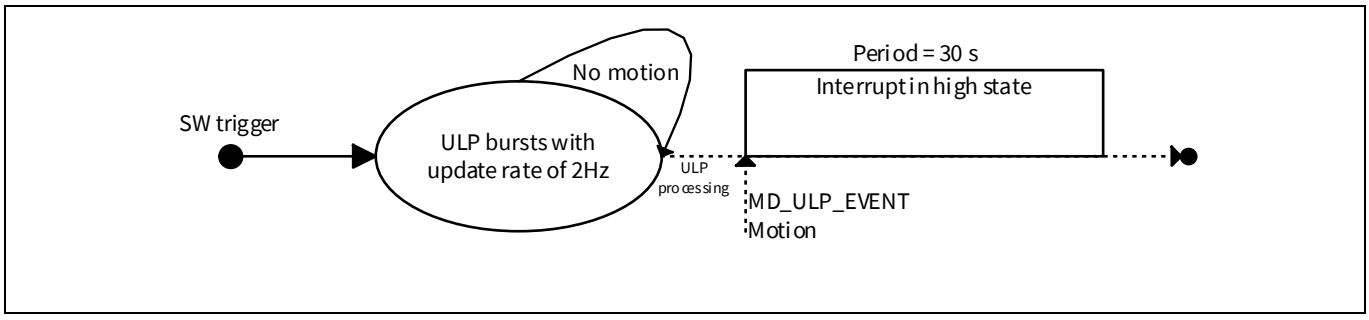


Figure 116 Configuration with ULP bursts

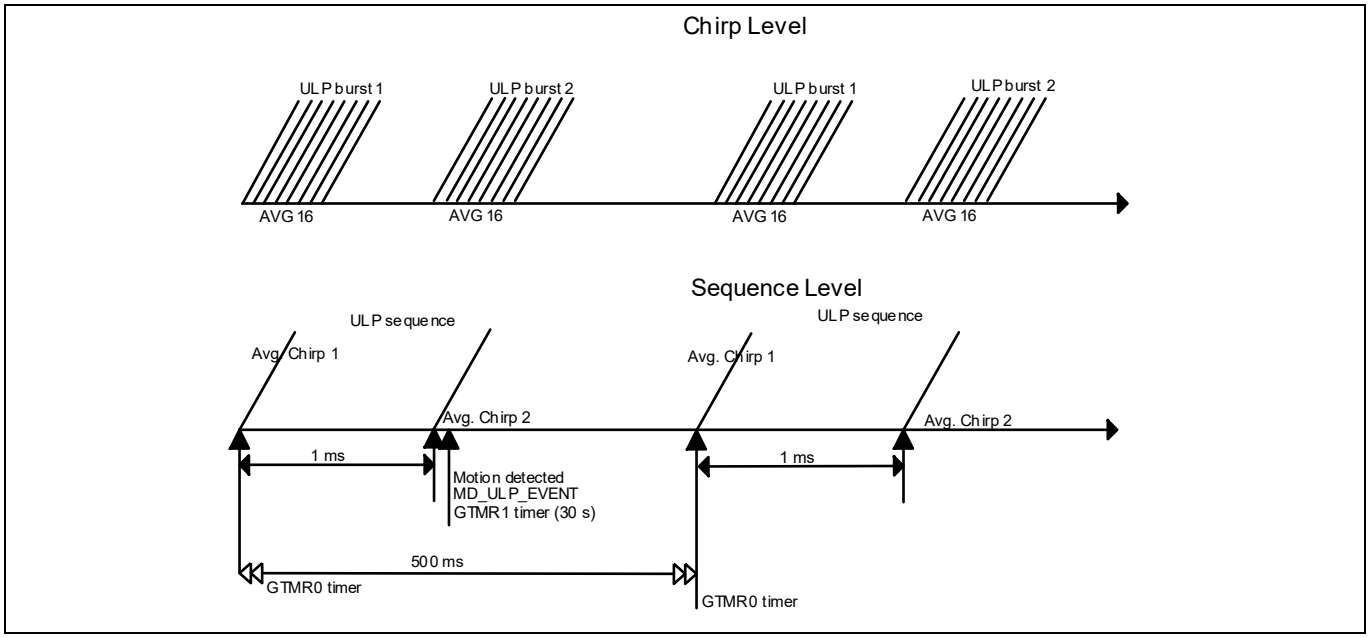


Figure 117 Timing model

Application use cases:

This configuration is particularly suitable as a standalone solution for smart appliances, such as lights, display controls for thermostats, or other similar devices. For example, after boot-up, the sensor continuously runs the ULP sequence. When motion is detected, the `MD_ULP_EVENT` interrupt is generated, holding its state for 30 seconds. This persistent interrupt along with a relay switch can be used to activate a connected smart appliance. This enables to perform the necessary tasks, such as turning on a light or displaying device information for a time interval of 30 seconds.

Power saving with an external microcontroller:

To further enhance power efficiency, this configuration can be controlled by an external microcontroller. In this setup, the sensor is programmed to operate ULP bursts at a 2 Hz update rate, but instead of running continuously, the microcontroller can externally trigger the sensor to execute two ULP bursts (spaced 1 ms apart) on demand.

After ULP processing is complete, if no motion is detected, the sensor enters an ultra-deep-sleep mode, further reducing power consumption. In such situations, the `MD_ULP_EVENT` interrupt's 30 second hold period is ignored, as the sensor transitions to its low-power state immediately after completing ULP processing. This capability makes the configuration even more efficient for battery-powered or energy-sensitive applications.

In summary, this highly flexible and power-optimized configuration can operate independently for smart appliances or integrate with an external microcontroller to substantially reduce power consumption in scenarios like lighting automation, thermostat control, or similar IoT applications. This ability to adapt between continuous operation and event-driven triggering offers versatility for a variety of use cases.

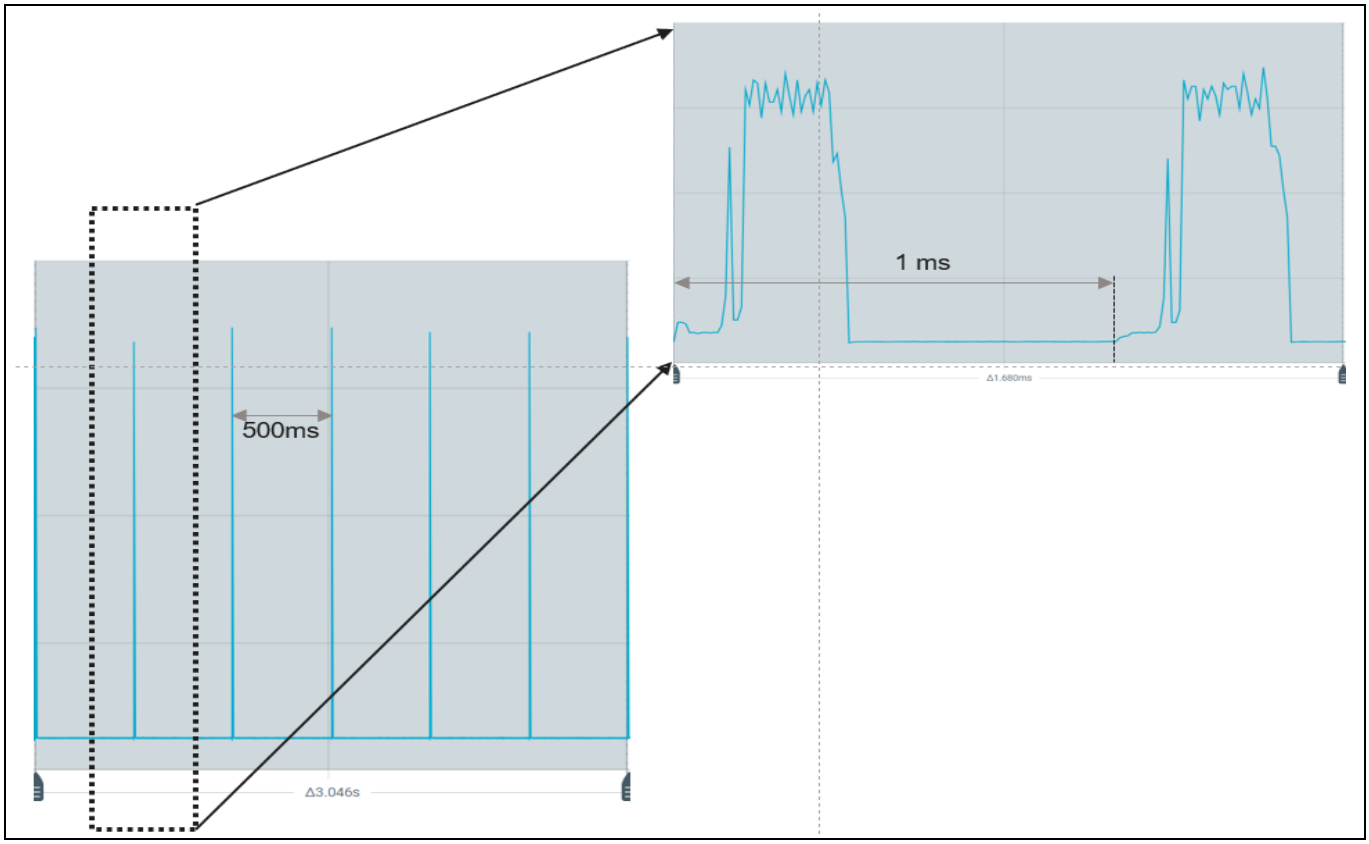


Figure 118 Nordic Power Profiler Kit showing the time interval of 500 ms between ULP sequences

Table 35 Power values

Voltage (V)	Current (mA)	Power (mW)
1.2	0.4	0.48
1.8	0.08	0.144
Total	–	0.62

The ULP sequence configuration, as shown in Figure 118, operates at a relatively low update rate of 2 Hz, as measured by the Power Profiler Kit. The figure also reveals the temporal relationship between consecutive ULP bursts, indicating that they are spaced 1 ms apart. The power consumption values associated with this ULP configuration are presented in Table 35, providing an approximation of the sequence's power efficiency.

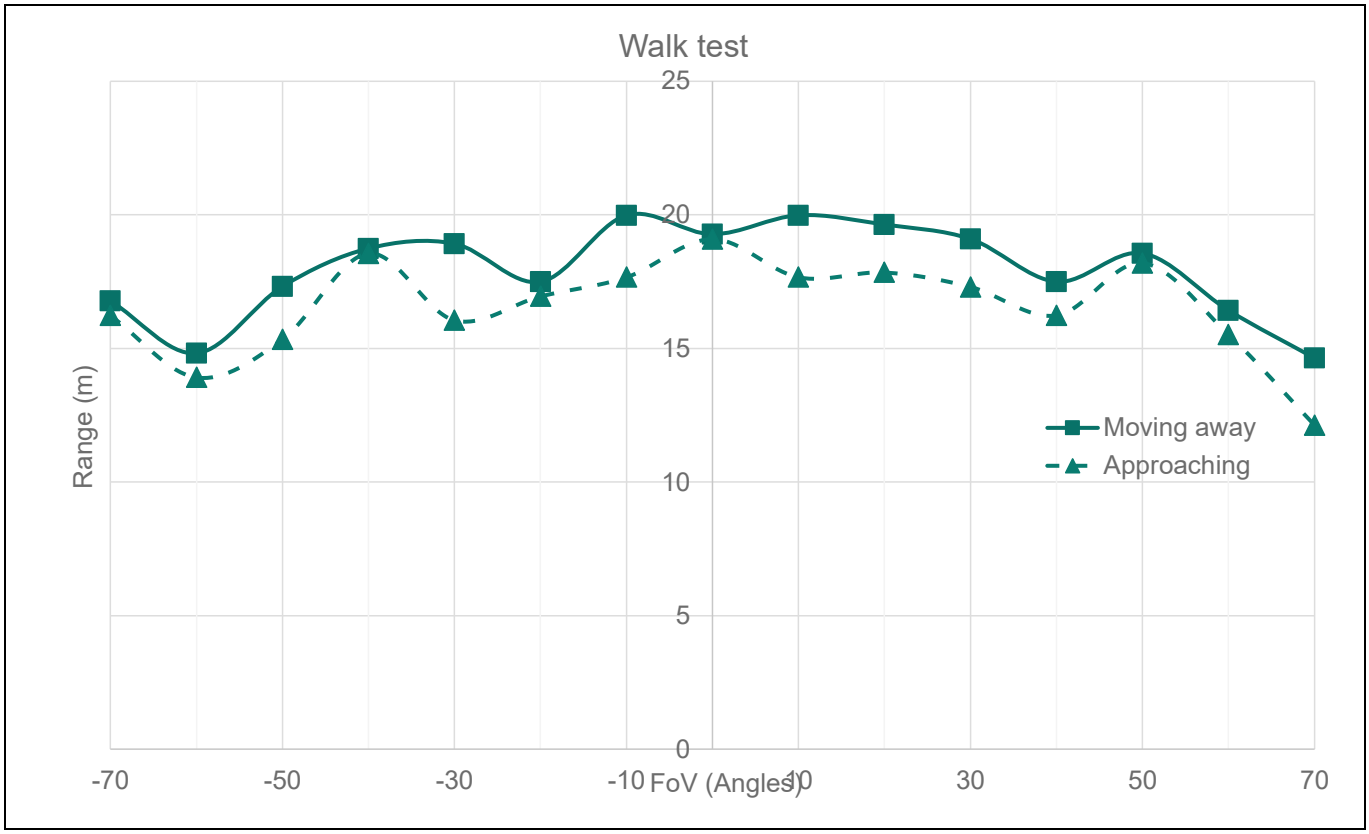


Figure 119 ULP target maximum range detections over the entire field of view in outdoor environment

Figure 119 presents the ultra-low power (ULP) target maximum range detections observed during the experiment. For this configuration, the results validate the sensor's maximum detectable range that can be achieved, when the target moves away and approaches the sensor across different angular sectors within the field of view. Since this configuration is utilizing ULP mode only, the Doppler masking feature is not available. This experiment highlights the effectiveness of the configuration in outdoor environments, even under complex and challenging conditions, ensuring precise and consistent motion detection over the specified range and angular coverage.

8.3.1 Operating modes

8.3.1.1 Default operating mode (auto interrupt clear)

This is the default operating mode of the configuration. The motion interrupt is passed to the host microcontroller or relay circuitry. If no motion is observed, the interrupt is automatically cleared after remaining in high state for 30 seconds. Figure 116 shows this behavior.

8.3.1.2 Hardware reset/power cycle (no SPI required)

The interrupt generated by the radar is cleared by the host microcontroller or relay using one of the following methods:

- Performing a hardware reset, or
- Powering OFF the radar upon receiving the interrupt, then powering it back ON after a defined time interval

No SPI communication is required for this mode of operation, as shown in Figure 113.

8.3.1.3 Motion-triggered power control (No SPI required)

When a motion interrupt is detected, the host microcontroller is responsible for the following:

- Powering ON the radar and triggering the range-Doppler sequence to confirm motion detection within a minimum time interval of 6 ms (SEQ1 active RF time plus MD processing time), or
- Powering OFF the radar if no motion interrupt is observed for the minimum time interval

No SPI communication is required for this mode of operation, as shown in Figure 120

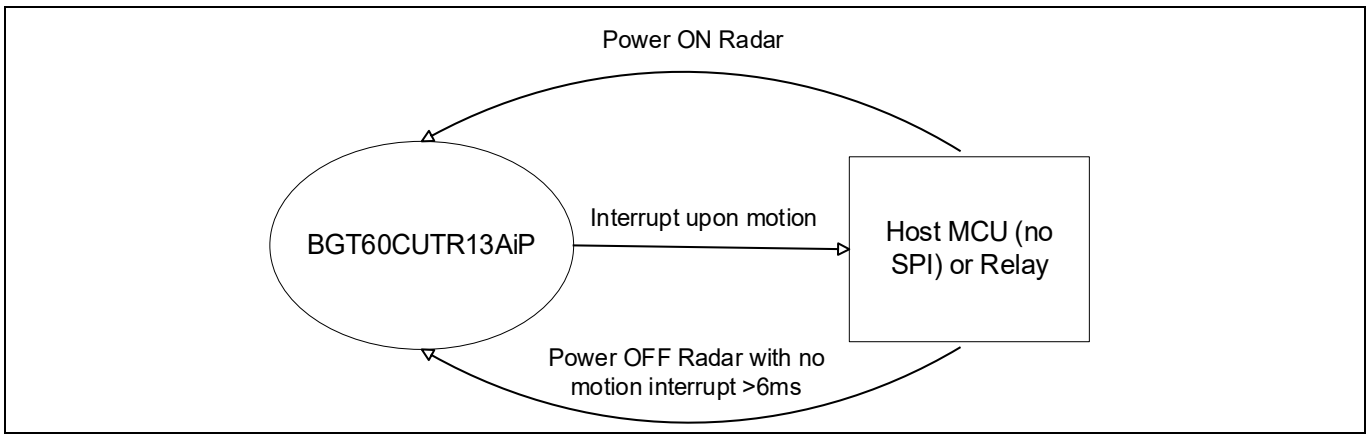


Figure 120 Motion interrupt-based power control with Host MCU

8.3.1.4 Interrupt clearing via SPI command (SPI required)

When a motion interrupt occurs, the host microcontroller or relay is responsible for clearing the interrupt by sending an SPI command.

This mode requires SPI communication between the host microcontroller and the radar, as shown in Figure 115.

9 PCB design guideline

9.1 XENSIV™ BGT60CUTR13AIP PCB schematics design guide

The XENSIV™ BGT60CUTR13AIP MMIC schematic can be optimized for a minimum bill of materials (BOM), requiring only the MMIC, a few bypass capacitors, and a crystal or oscillator (40/80 MHz). The minimum interface configuration includes a single SPI and one IRQ (Interrupt towards SOC), which is typically assigned to DIO4, but can also be assigned to other DIOs.

The MMIC provides several test options, including:

- OSCCLK, which can provide the internal oscillator signal (40 MHz or 80 MHz)
- DIV, which can be enabled to provide the VCO divided output signal (1/16)

Handling unused pins:
To prevent interference with the MMIC's operation, any unused pins should be left unconnected (N.C.), or floating. This will ensure that the MMIC operates as intended and minimizes the risk of errors or malfunctions.

9.2 XENSIV™ BGT60CUTR13AIP pin assignment for minimum configuration

The XENSIV™ BGT60CUTR13AIP device also provides several DIO pins, which can be used for various purposes. Table 36 summarizes the available signal pins:

Table 36 Pin assignment for minimum configuration

DIO Pin	Description	Minimum configuration	Remark
RST_N	HW reset pin	Input	See Table 2
CLK	SPI, clock	Input	SPI_CLK
DIO0,	SPI, data in	Input	SPI_MOSI
DIO1	SPI, data out	Output	SPI_MISO
CS_N	SPI, Chip select	Input	SPI CSN
DIO4	Infineon proposal: FIFO interrupt or MD interrupt or both	Output (also Input)	see Table 2
DIO5	Infineon proposal: secondary MMIC interrupt or DMUX e.g. as motion detection interrupt	Output (also Input)	see Table 2
XI	Connected to XTAL	Input	2nd function as oscillator input, see Table 2
XO	Connected to XTAL	Output	
DIO2	Quad SPI or bootloader selection	N.C.	See Table 2
DIO3	Quad SPI or bootloader selection, optional trigger input	N.C.	See Table 2
CAL	For Infineon production testing only	N.C.	

DIO Pin	Description	Minimum configuration	Remark
OSCCLK	Oscillator output signal	N.C.	Enabled by registers See Table 2
DIV	RF divider output	N.C.	Enabled by registers See Table 2

9.2.1 XENSIV™ BGT60CUTR13AIP schematic for minimal configuration

Figure 121 gives an overview of the pin assignment for the XENSIV™ BGT60CUTR13AIP MMIC, highlighting the connections for digital signals, supply, and ground. The schematic shows the minimal configuration required for operation, with unused digital signals left floating (N.C.). Note that VSSRF can be connected together for improved grounding. Additionally, NC pins A1 and M1 are floating pins, but can be connected to GND if the PCB design requires a better GND distribution to optimize performance.

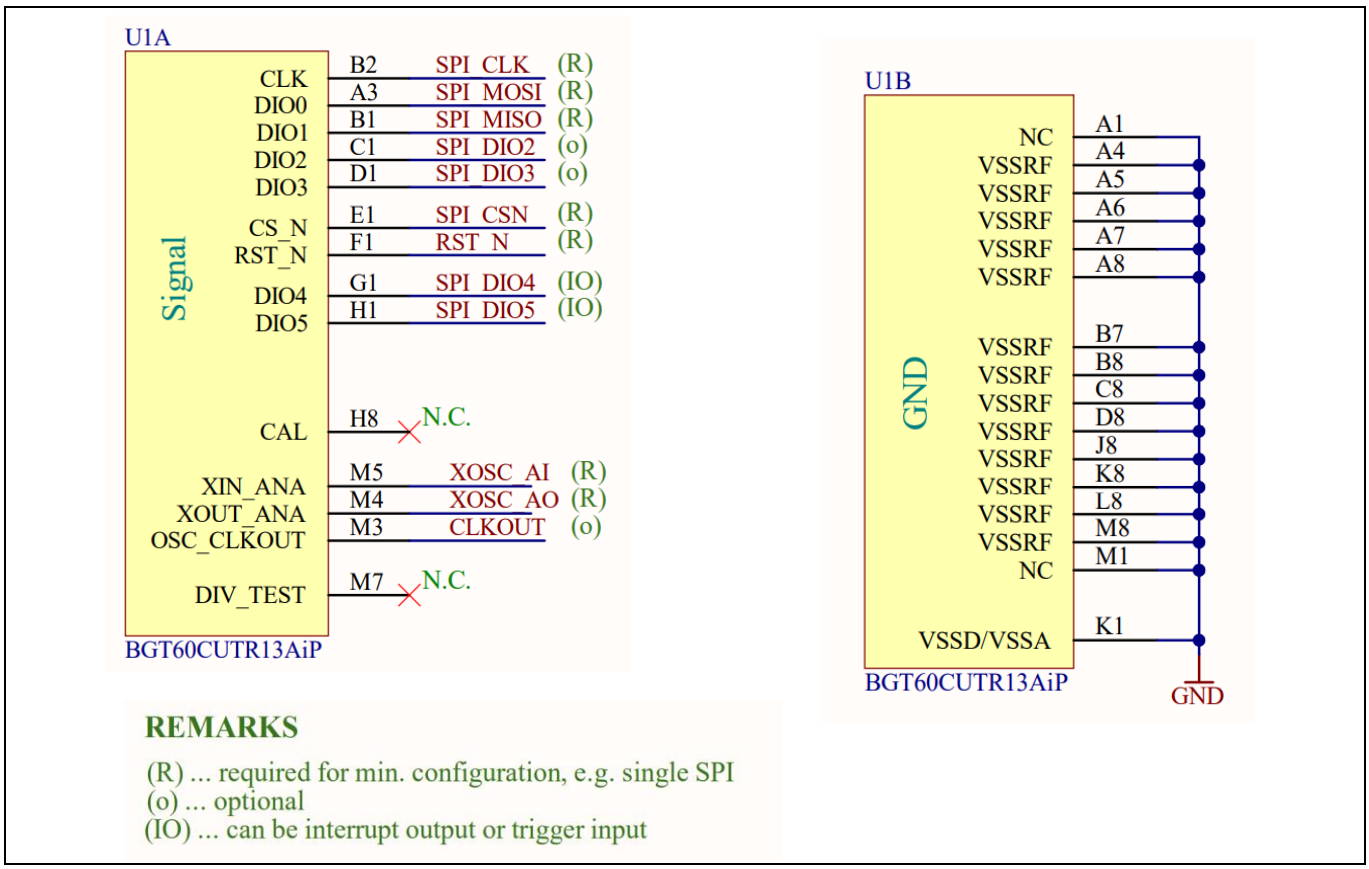


Figure 121 Pin assignment and schematic overview for minimal configuration

9.2.2 XENSIV™ BGT60CUTR13AIP supply domains and bypass capacitors

The XENSIV™ BGT60CUTR13AIP MMIC has improved sensitivity to spurious signals compared to BiCMOS sensors, allowing for a more robust design. Additionally, the MMIC itself features several on-chip low dropout (LDO) regulators with high power supply rejection ratio (PSRR), which helps to reduce the supply filtering effort and relaxes the LDO noise requirements. This built-in PSRR provides an additional layer of noise suppression, making the design more robust and easier to implement.

To minimize the bill of materials (BOM), only the minimum number of capacitors required by the LDO specification is necessary. In fact, only one bypass capacitor per MMIC supply domain is expected to be sufficient.

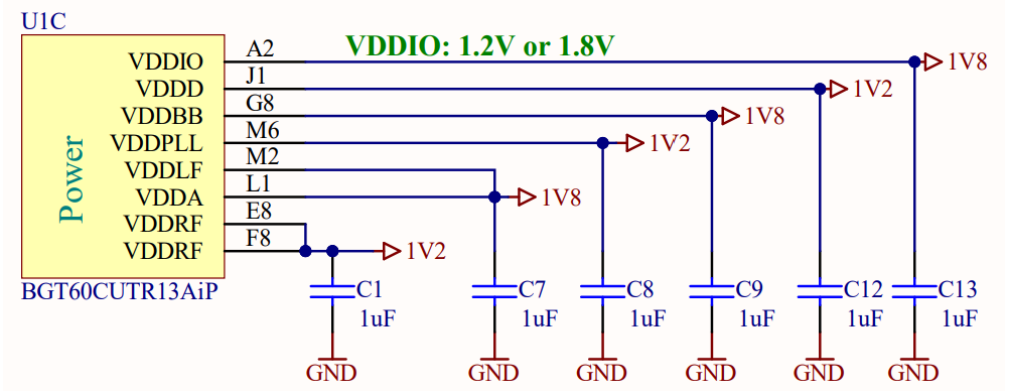


Figure 122 Power supply domains

Maximum currents for LDO/DCDC selection:

To give you an idea of the maximum currents required for the LDOs, here are the maximum currents for the 1.2 V and 1.8 V domains:

- 1.2 V domains with a maximum total current below 500 mA
- 1.8 V domains with a maximum total current below 150 mA

Note: The maximum currents listed are the maximum expected currents for each domain. The actual currents may be lower, depending on the specific application and operating conditions. Please refer to the latest datasheet for the most up-to-date and accurate information.

The following Table 37 provides an overview of the supply domains for the XENSIV™ BGT60CUTR13AIP, including the minimum configuration requirements and maximum current expectations.

Table 37 Supply domains

Supply Pin	Description	Maximum currents	Remark
VDDIO	Digital I/O supply	< 1 mA	VDDIO is compatible with both 1.8 V and 1.2 V supplies; digital signals will match the I/O signal level accordingly See [1]
VDDD	Digital core supply	< 9 mA	See [1]
VDDBB	Baseband supply	< 77 mA	More sensitive, 1 µF bypass cap
VDDPLL	PLL supply	< 108 mA	See [1]
VDDLF	LF supply	< 14 mA	More sensitive, 1 µF bypass cap
VDDA	Analog supply	< 16 mA	See [1]
VDDRF	RF supply	< 303 mA	More sensitive, 1 µF bypass cap

[1] The 1 µF bypass capacitor is recommended for VDDRF, VDDLF, and VDDBB to ensure stable operation, while 100 nF is sufficient for the other supply domains.

Recommended bypass capacitor specifications

For optimal performance, it is recommended to place the bypass capacitors close to the MMIC, to minimize parasitic inductances and ensure effective decoupling. The following specifications are recommended for the MMIC bypass capacitors:

- Type: multilayer ceramic capacitors (MLCC), X5R dielectric
- Tolerance: ±20%
- Rated voltage: ≥ 6.3 V
- Size: 0201 or 0402

The on-chip LDOs with high PSRR, combined with the external bypass capacitors, help to improve the supply noise profile and reduce the external filtering effort.

9.3 External clocks

The XENSIV™ BGT60CUTR13AIP MMIC requires an external clock source to operate. Infineon has tested and recommends several crystal (XTAL) options for use with the MMIC, which have shown good performance.

External crystal clock schematic

Figure 123 shows the recommended crystal schematics, including load capacitors C10 and C11, for both 40 MHz and 80 MHz crystal options.

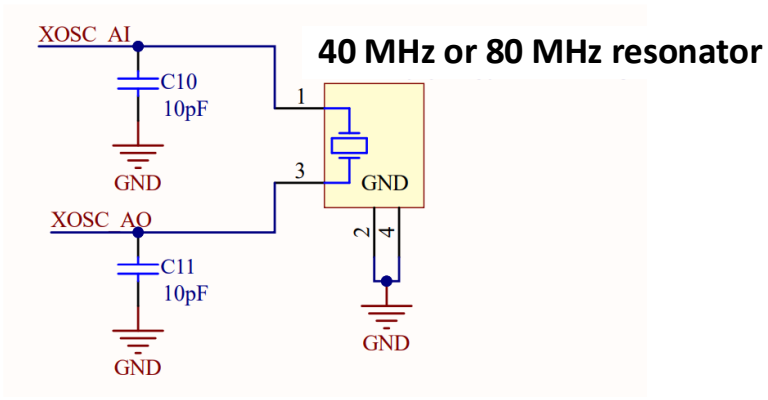


Figure 123 Typical crystal resonator circuit with load capacitors

Note: For a cost-optimized Bill of Materials (BOM), crystal resonators have an advantage compared to oscillators, as they are generally more cost-effective while still providing reliable performance. Infineon recommends using crystal resonators as the external clock source for the BGT60CUTR13AIP MMIC.

Load capacitors selection
To improve the overall system performance, it is recommended to use high-quality crystal load capacitors. The following specifications are recommended:

- Type: NP0 (COG) capacitors for better temperature stability
- Tolerance: ±2% (as recommended by the supplier)
- Rated voltage: ≥ 6.3 V
- Size: 0201 or 0402 (as commonly used in electronic designs)

Using NP0 (COG) capacitors with the specified tolerance, rated voltage, and size can help to minimize temperature-related drift and ensure stable clock signal performance. This, in turn, can improve the overall system performance and reliability of the XENSIV™ BGT60CUTR13AIP MMIC.

Proposal for external clocks
The following XTAL options are recommended:

9.3.1 External 80 MHz XTAL:

Recommended series resistance: ESR up to 50 Ω, with best performance at ESR = 30 Ω

Example XTALs:

- NDK, NX2016SA 80.000MHz (NX2016SA-80M-STD-CZS-3)
- KDS, PN#7AF08000A09, size: 2016 (Infineon reference number)
- Hosonic, PN# E1SB80E001U00E, size: 2016
- TXC, 8Y80062002, size: 2016

9.3.2 External 40 MHz XTAL:

Recommended series resistance: ESR up to 50 Ω

Example XTALs:

- NDK, NX2016SA 40.000MHz (NX2016SA-40M-STD-CZS-3)
- KDS, PN#7AF04000A4C, size: 2016 (Infineon reference number)
- Hosonic, PN#E1SB40E001U00E, size: 2016
- TXC, 8Y40062009, size: 2016

9.3.3 XTAL Remarks

When selecting an XTAL, the following should be considered:

- Recommended series resistance: better or equal to 50 Ω
- Temperature range: according to the product temperature range and product requirements
- Capacitive load: 8 pF to 12 pF, including PCB traces (result from simulation)
- Frequency tolerance: ±10 ppm (up to ±20 ppm for cost advantages)
- Frequency accuracy vs. temperature:
  - –40°C to 85°C: 20 ppm
  - –40°C to 105°C/125°C: up to ±50 ppm (relaxed values)
- XTAL temperature stability: capacitors with NP0 (COG) dielectric are recommended for better ppm values

Note: The frequency tolerance and accuracy values are specified to ensure optimal performance of the MMIC. However, for cost advantages, a relaxed frequency tolerance of ±20 ppm may be acceptable.

9.4 External MMIC power supply

The external supply is a critical component of the Infineon XENSIV™ BGT60CUTR13AIP radar IC, requiring careful consideration to ensure proper operation. The key requirements include the use of low dropout (LDO) regulators, specifically 1.2 V and 1.8 V LDOs, with precise power-up sequencing to prevent potential issues. Alternatively, 1.2 V DC-DC converters can also be used, offering a viable option for powering the 1.2 V domain.

9.4.1 Supply noise requirements: maximum spurious signal levels

This section provides an overview of the key requirements and considerations for the external power supply, including the use of low dropout (LDO) regulators, DC-DC converters, and supply filtering. The following topics are covered:

- LDO schematics and power-up sequence
- LDOs and DC-DC converters for MMIC supply
- Supply power-up sequence and timing
- Supply filtering and capacitor selection
- Considerations for radiated and conducted emissions

By following the guidelines and recommendations outlined in this section, designers can ensure that their external power supply meets the requirements of the XENSIV™ BGT60CUTR13AIP radar IC and provides optimal performance and reliability.

Determining maximum allowable spurious signal levels

To find the maximum allowable spurious signal levels for the XENSIV™ BGT60CUTR13AIP MMIC supply domains, we performed a series of tests. Here is a step-by-step overview of the test sequence:

1. **Inject a test tone:** Introduce a constant test tone with adjustable amplitude into the system
2. **Measure the range-Doppler map:** Process the signal using 64 samples and 64 chirps, and then measure the resulting range-Doppler map
3. **Increase the test tone amplitude:** Gradually increase the amplitude of the test tone until the spurious tone becomes visible, approximately 3 dB above the range-Doppler noise floor
4. **Repeat for multiple test tones:** Repeat Step 1 to Step 3 for multiple test tones to ensure a comprehensive understanding of the system's behavior
5. **Test multiple supply domains:** Repeat the entire process for each supply domain (e.g., 1.2 V and 1.8 V), while maintaining the other supply domains at a normal low-noise supply level

By following this test sequence, you can determine the maximum allowable spurious signal levels for your specific application and ensure optimal system performance.

Interpreting the results and applying them to your system

The results of our investigation are summarized in Figure 124, which shows the optimal supply noise levels that do not visibly impact the range-Doppler domain. To ensure proper sensor performance, it is essential to limit the maximum spurious signal levels for all supply domains.

How to read the results?

Figure 124 shows the maximum spurious signal levels for the 1.2 V and 1.8 V domains, measured over a frequency range of 100 kHz to 6 MHz. The test tone levels that generate a –85 dBFS dynamic range at the range-Doppler map were used to determine these levels.

Key results

The maximum spurious signal levels for the 1.2 V and 1.8 V domains are as follows:

- All 1.2 V domains combined: < –60 dBm, equivalent to approximately 600 µVpp or 200 µVrms
- All 1.8 V domains combined: < –55 dBm, equivalent to approximately 1.1 mVpp or 380 µVrms
- Noise levels below 100 kHz or above anti-aliasing frequency are not critical and can be higher as the ones specified above

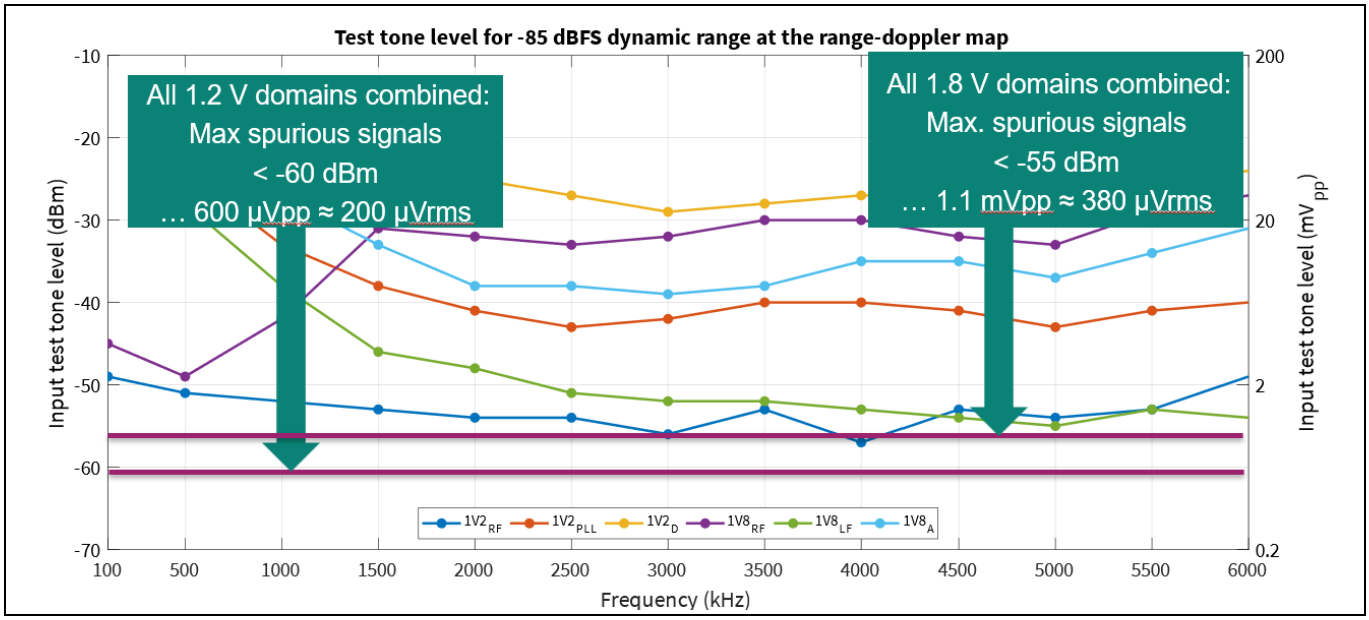


Figure 124 Maximum spurious signal levels for 1.2 V and 1.8 V domains, measured from 100 kHz to 6 MHz

Applying the results to your system

To ensure optimal system performance, it is crucial to ensure that the supply noise levels in your system remain within these specified limits. Exceeding these maximum spurious signal levels can compromise sensor performance and dynamic range.

What to do next?

1. Review your system's supply noise levels to ensure they are within the specified limits
2. If your system's supply noise levels exceed the specified limits, take steps to reduce the noise levels, such as using noise-reducing components or optimizing your system's design
3. Verify that your system's performance meets the required specifications after implementing any changes to reduce supply noise levels

By following these steps and ensuring that your system's supply noise levels remain within the specified limits, you can maintain optimal system performance and ensure reliable operation.

9.4.2 LDO schematics

Figure 125 depicts the power-up sequence for the 1.2 V and 1.8 V LDOs, including an enable delay between the two domains. The delay, implemented using the RC delay components R1 and C6, ensures that the 1.2 V domain is powered up first, allowing for proper configuration of the MMIC's internal level translators and preventing potential issues.

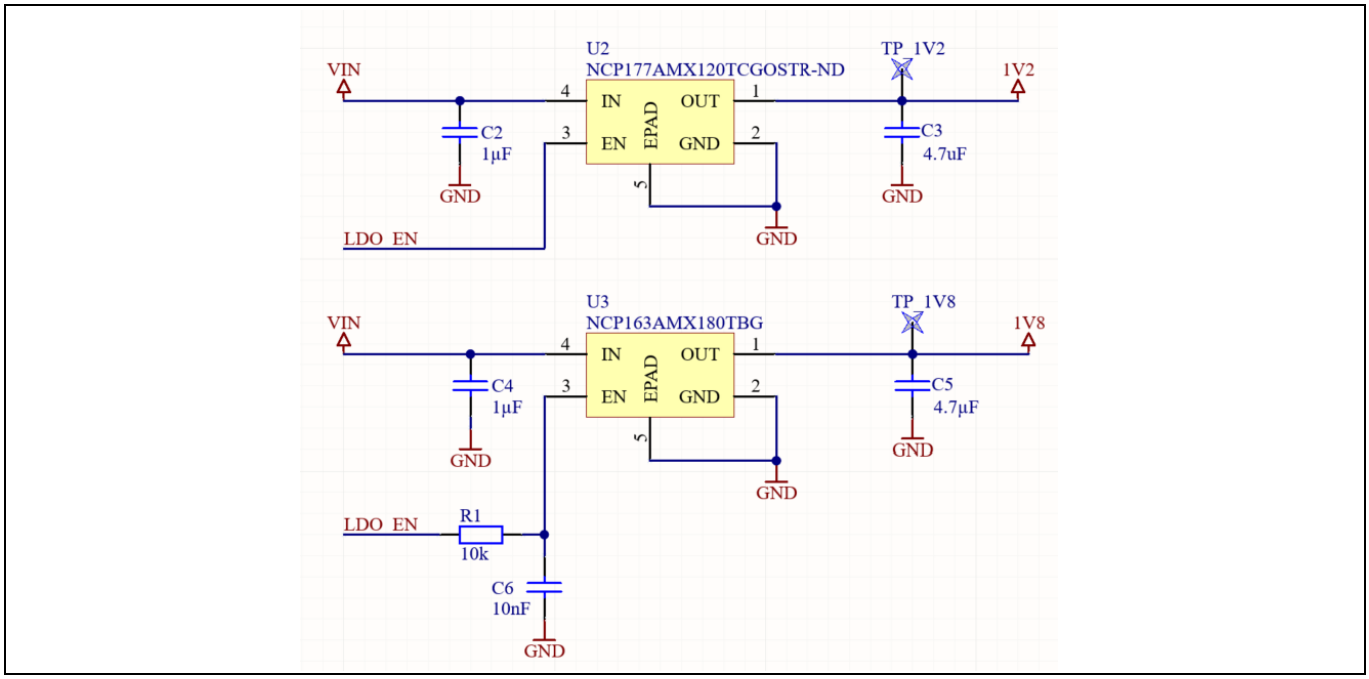


Figure 125 LDO configuration for 1.2 V and 1.8 V supply

9.4.3 LDOs (low-dropout regulators) and DC-DC converters

MMIC supply requirements

The MMIC supply requires two regulators to provide stable voltage levels for proper operation. The general requirements are as follows:

- 1.2 V: minimum 500 mA
- 1.8 V: minimum 150 mA

It is essential that the voltage levels remain within the datasheet specified values of ±5% during operation.

LDO suppliers

Low-dropout (LDO) regulators are a suitable option for the MMIC supply.

onsemi

Following were tested on Infineon EVKs:

- NCP177 (onsemi), 500 mA, fast transient LDO, available in a 1.2 V variant
- NCP163 (onsemi), 250 mA, low noise LDO, available in a 1.8 V variant

Toshiba

- TCR3DM, 300 mA
- TCR5BM12, 500 mA, good PSRR, different pin assignment

Texas Instruments

- TLV740P, 300 mA, LDO regulator with foldback current limit, for 1.8 V
- TLV755P, 500 mA, high-PSRR low-IQ low dropout voltage regulator with enable, for 1.2 V

Note that other LDOs with similar performance characteristics can also be used, provided they meet the required specifications. To determine suitability, compare the datasheets of alternative LDOs, and ensure that they offer comparable noise performance, current rating, and voltage regulation.

DC-DC converters

As an alternative to LDOs, DC-DC converters can be used to power the MMIC, offering a viable option for powering the device. However, to prevent the introduction of spurious signals in the IF domain, it is essential to ensure that the DC-DC switching frequency exceeds the MMIC's anti-aliasing filter frequency.

The following DC-DC converters have been tested at Infineon and meet the requirements:

- TPS62171DSGR (Texas Instruments), 2.25 MHz, 500 mA, MMIC anti-aliasing filter frequency shall be below 2.25 MHz
- TPS62808YKAR (Texas Instruments), 1.5 MHz, 600 mA, MMIC anti-aliasing filter frequency shall be below 1.5 MHz
- RP509Z182B-E2-F (Nisshinbo), 6 MHz, 500 mA, MMIC anti-aliasing filter frequency shall be below 6 MHz

These DC-DC converters have been tested under the following conditions:

- No DC-DC spurious signals visible in the Range-Doppler (RD) map
- Bare minimum number of capacitors used
- High-pass filter (HPF) frequency: 350 kHz
- Anti-aliasing filter (AAF) frequency: 2 MHz

Additionally, the MMIC's internal LDO power supply rejection ratio (PSRR) will effectively suppress low-frequency noise below the baseband high-pass filter frequencies, providing an additional layer of noise reduction and helping to maintain the overall signal integrity.

9.4.4 Supply power-up sequence

To prevent possible low leakage currents within the MMIC, a specific power-up sequence must be followed:

- First, the 1.2 V domain must be powered up
- After a delay of 20 µs, the 1.8 V domain can be powered up

This delay is crucial to ensure that the digital supply is powered up first, allowing the MMIC's internal level translators to configure correctly and prevent leakage from the 1.8 V to the 1.2 V domain.

The power-up sequence can be controlled by either the system-on-chip (SoC) or the host processor, which can manage the power-up sequence and ensure that the delay is implemented correctly. Alternatively, an RC-time delay can be implemented in hardware directly, using a resistor and capacitor to create a delay circuit that meets the required 20 µs delay.

See Section “Power supply startup” for more information on power supply startup procedures.

9.4.5 Supply filtering

The supply filtering section is critical in ensuring the stability and performance of the XENSIV™ BGT60CUTR13AIP MMIC. The choice of filter components, including capacitors and ferrite beads, depends on the main system power supply and the specific requirements of the application.

General guidelines and considerations

When selecting filter components, consider the following:

- The minimum bypass capacitors for the LDOs are specified by the LDO supplier and are required for LDO stability
- Filter components, including capacitors and ferrite beads, should be selected based on the specific requirements of the application and the main system power supply
- Supply filtering should be done before the LDO to minimize voltage drop and ensure stable operation
- If a ferrite bead is required, it should only be placed on the LDO input side to avoid voltage drop after the LDOs
- By verifying the supply noise behavior as described in Section Supply noise requirements: maximum spurious signal levels and shown in Figure 124, additional filtering can be applied to further optimize the supply filtering, if necessary
- Consider the following capacitor values:
  - Minimum input filter capacitors: > 1 µF
  - Minimum output filter capacitors: > 4.7 µF
- Consider the supply noise behavior when selecting capacitors, including the frequency range and amplitude of the noise

Additional considerations for radiated and conducted emissions

In addition to the general guidelines, supply filtering should also consider rules for radiated and conducted emissions, which are specific to the application and must be considered on a system level. The system integrator must verify the radiated and conducted emissions and perform the required actions to ensure compliance with relevant regulations and standards.

10 RF shield design, simulation, and measurement

This chapter is intended to provide shield board design guidelines for Infineon 60 GHz radar sensor users.

There are major factors to consider when developing a radar shield board:

- The required field of view (FoV)
- The space allocated to the radar board
- The accepted limits of angle error in radar mode (in case of motion detection)

RF antenna designers typically focus on optimizing antenna gain, impedance matching, and the phase linearity of the radiated electric field. Electromagnetic (EM) simulations are commonly performed in the frequency domain, at a single frequency or over a narrow frequency sweep, with the antenna excited in radiation mode. FMCW radar operation, however, involves time-dependent behavior that is not captured by these radiation-mode simulations alone. Without a dedicated radar-mode simulation, signal-magnitude dips at specific azimuth angles and the resulting reduction in the usable field of view (FoV) may remain undetected until anechoic-chamber measurements are performed on the fabricated and assembled device. Identifying these issues at such a late stage can necessitate a costly and time-consuming redesign and fabrication cycle for the RF shield board.

Note: Infineon's updated design flow includes radar-mode simulations. This allows the designer to compute the signal magnitude and the angular error across the entire field of view before tape-out and to verify that the predicted radar performance meets the specifications.

The flow, the underlying methodology and the resulting design conclusions are described in the following sections. The XENSIV™ BGT60CUTR13AIP sensor is used as the device under test throughout this document.

10.1 Radar mode simulation setup

The radar-mode simulation flow uses ANSYS HFSS for EM analysis, Python and MATLAB for data post-processing. The flow consists of three steps:

1. **Far-field pattern computation:** The far-field pattern of each antenna element is computed across the radar bandwidth using an interpolating frequency sweep
2. **Radar matrix computation:** The far-field data is linked to a shooting-and-bouncing-ray (SBR+) setup. A target is placed at a fixed distance from the radar and rotated through a parametric azimuth sweep covering the entire field of view, yielding the composite TX/RX signal at every spatial point
3. **Data post-processing:** The radar matrix is exported and processed in MATLAB to extract the angle of arrival (AoA), the angular error and the TX/RX magnitude versus azimuth

These three steps make it possible to characterize the radar response of a complete shield-board design before fabrication.

10.1.1 Far field computation

The antenna far field is computed through an EM simulation of the laminate on the shield board in radiation mode shown in Figure 126. An interpolating frequency sweep is defined for example from 57.5 GHz to 62 GHz. The boundary condition is defined as either an open region or a radiation box.

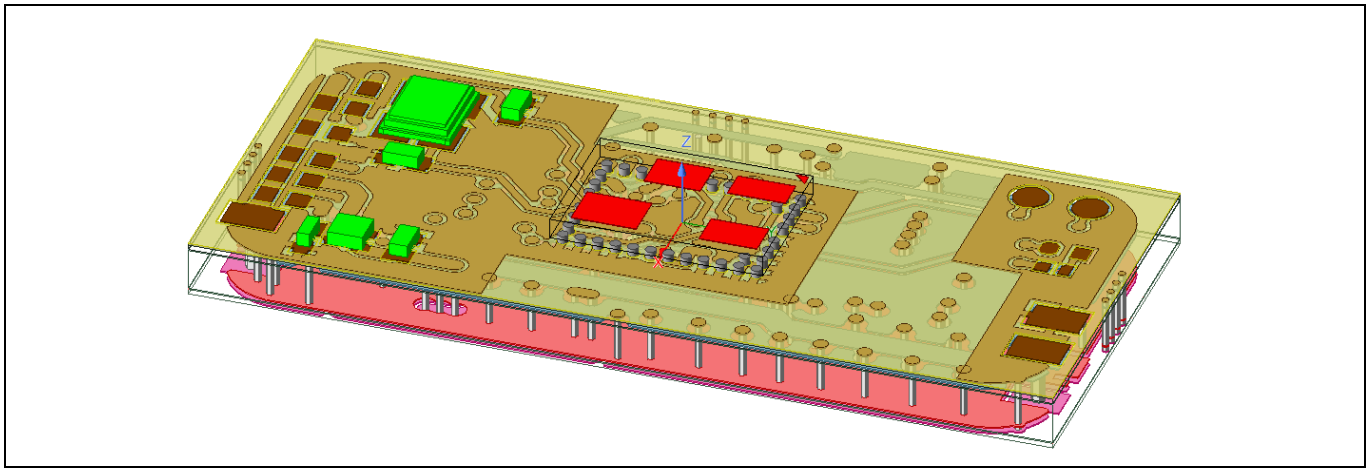


Figure 126 HFSS model of laminate package on RF shield board in radiation-mode EM simulation

It is important to define the far-field phase center for each antenna terminal. For a microstrip patch, the phase center is usually defined at the geometrical center of the patch.

10.1.2 Radar matrix computation

In the shooting and bouncing ray (SBR+) radar mode simulation setup, the radar matrix is computed at the different spatial points where a conical target is defined. Figure 127 shows SBR+ simulation setup including laminate package on RF shield board at the center of the coordinate system and a corner reflector at 1.75 m. In a 1TX 3RX radar operation, the radar matrix consists of a 3x1 matrix. Each element is the composite TX/RX signal after hitting the target. In addition to the corner reflector, the top metal of the PCB shield and the baseboard outline together with the corresponding components are added as PerfE reflectors. This addition corresponds to the case when the radar PCB is mounted on a carrier board. The top metal layer and its SMD components behave like parasitic reflectors, therefore it results in back-and-forth reflections between the target and the complete radar device.

Note: HFSS and SBR+ are registered trademarks of ANSYS, Inc. The use of HFSS/SBR+ simulation results in this document does not imply any endorsement or affiliation with ANSYS, Inc. All intellectual property rights related to the HFSS/SBR+ software, and its underlying technology are owned by ANSYS, Inc.

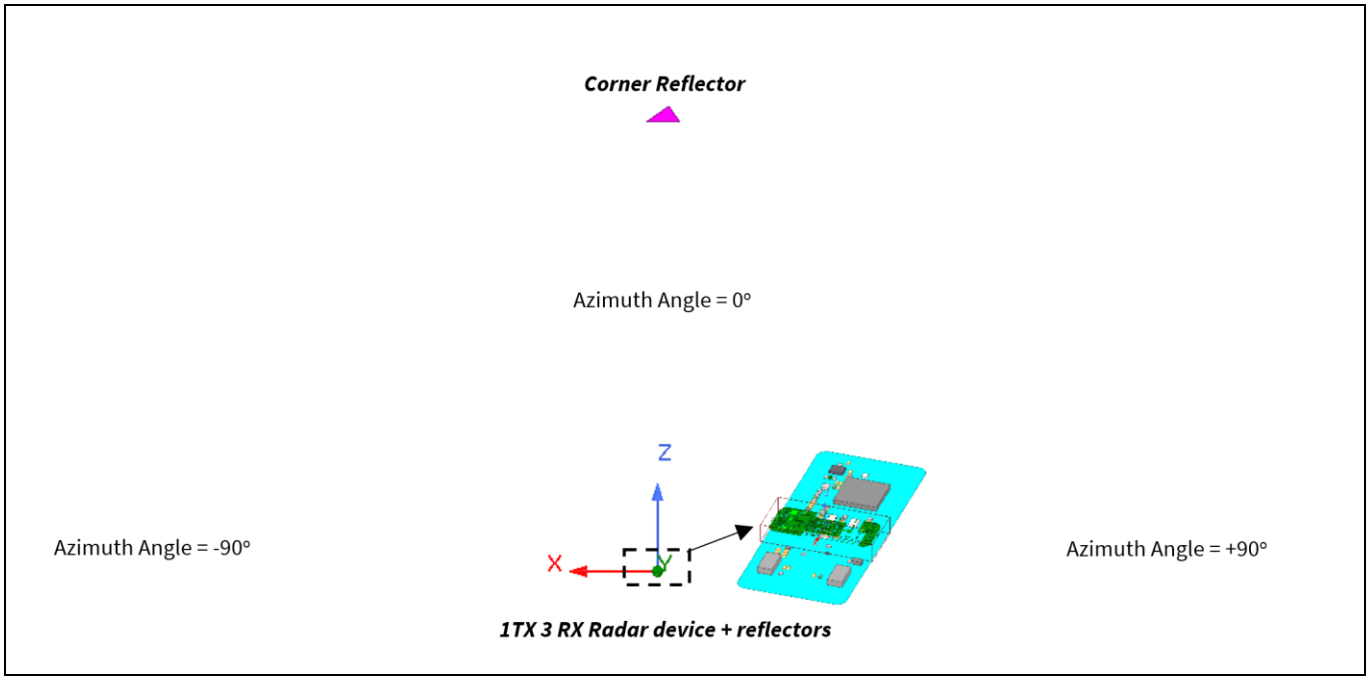


Figure 127 SBR+ simulation setup including laminate package on RF shield board at the center of the coordinate system and a corner reflector at 1.75 m for radar mode simulation

The FMCW radar analysis is defined in SBR+ mode according to the radar configurations in Figure 128. The target is rotated from –90° to +90° in the time from 0 second to 1 second, resulting in 181 3 × 1 matrix corresponding to the composite TX/RX signal reflected from the target at a given Azimuth angle. Figure 129 shows the magnitude of each TX/RX signal in the time (from 0 second to 1 second) and frequency (from 57.5 GHz to 62 GHz) domain. The magnitude and phase of each TX/RX signal depend on the back-and-forth reflections between the target at a specific location and the radar PCB. Figure 130 presents a visual ray tracing for these transmitted and reflected signals at t = 0.5 seconds equivalent to Azimuth Angle = 0°. To accurately simulate this mechanism, critical parasitic reflectors around the radar device should be considered as PerfE metal sheets in SBR+ simulation. These reflectors are usually presented by the top metal layer of the shield PCB, the top surface components around the radar device, the base microcontroller (MCU) board outline, and the corresponding top surface components.

This set of 3 × 1 matrix data is then exported to MATLAB for data processing and extraction of the angle of arrival and the angular error using mutual coupling calibration.

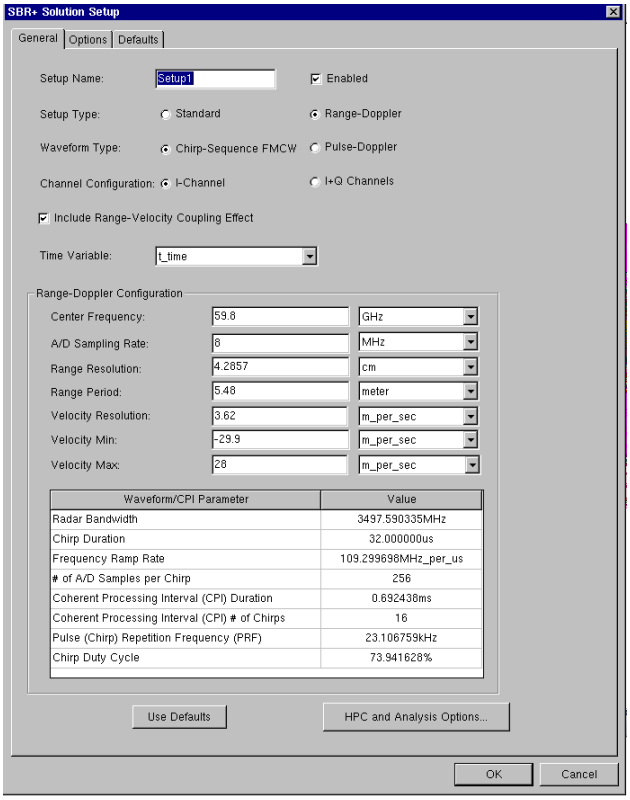


Figure 128 The 3D output matrix (shown in Figure 129) in the time domain from 0 second to 1 second and in the frequency domain from 57.95 GHz to 61.52 GHz equivalent to ~ 3.5 GHz radar bandwidth with a center frequency at 59.8 GHz

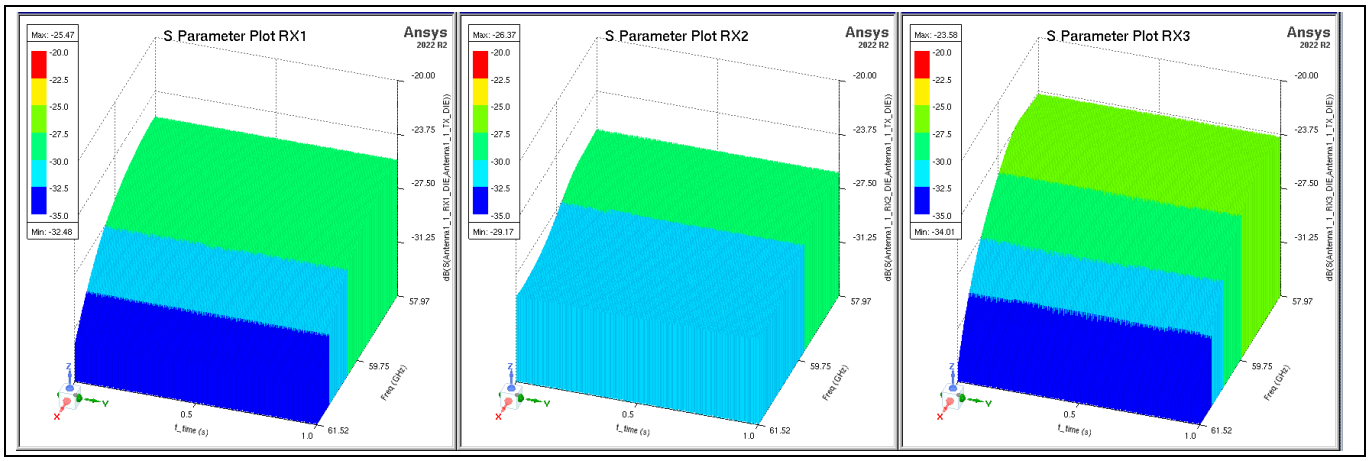


Figure 129 SBR+ 3D output matrix from simulation

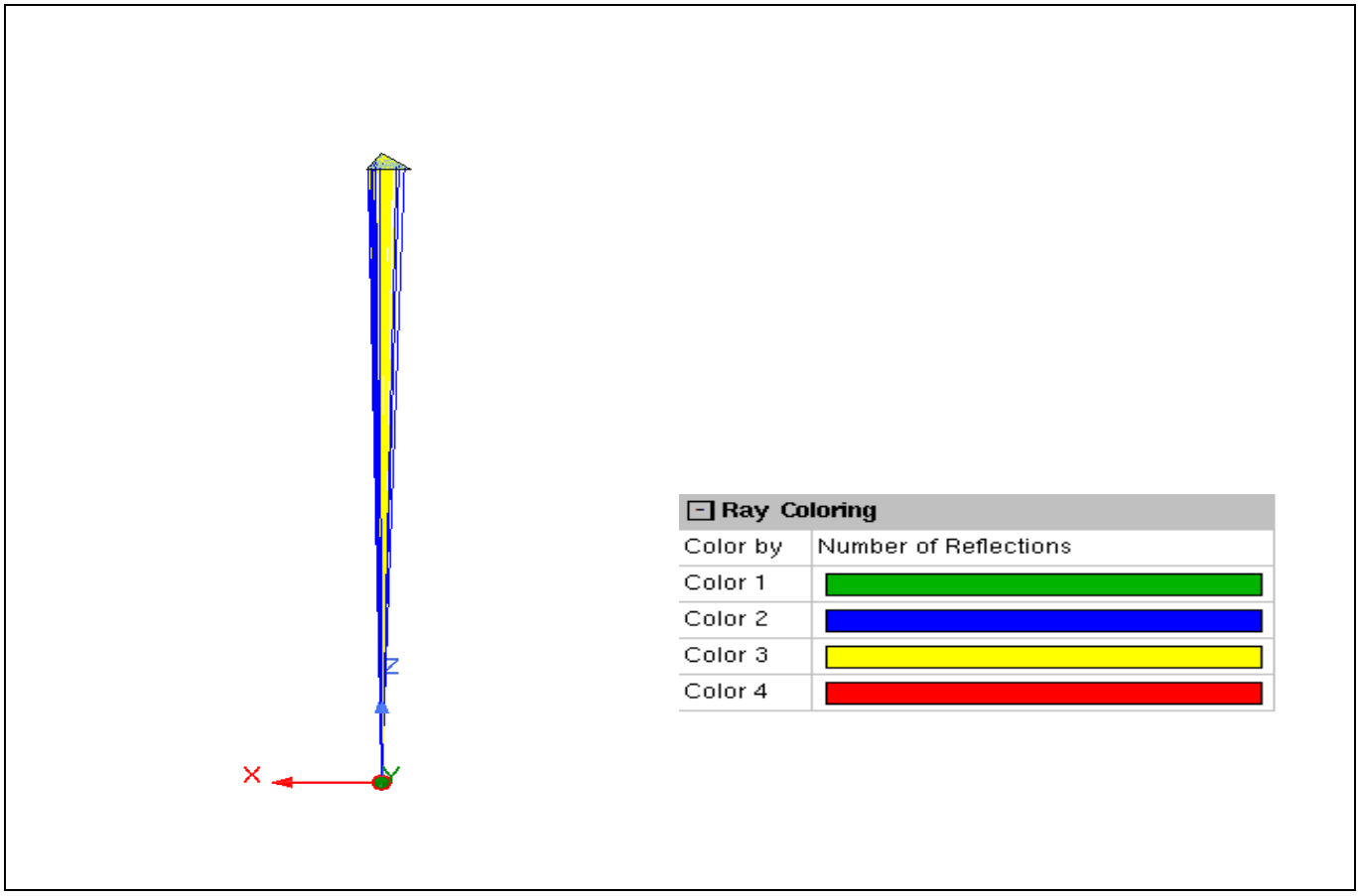


Figure 130 Visual ray tracing showing the transmitted and multiple reflected signals at t = 0.5 second corresponding to a target at azimuth angle = 0°

10.1.3 Data post-processing

Figure 131 shows the top-level data pipeline view based on the data from SBR+ simulation platform and the data coming from the anechoic chamber.

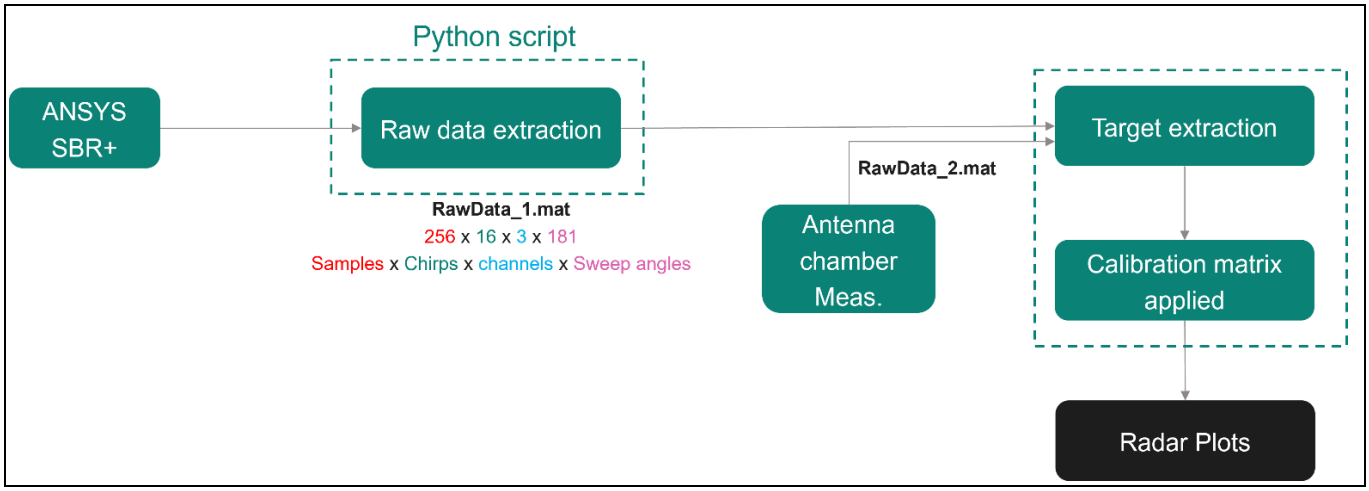


Figure 131 Top-level data pipeline view based on data from ANSYS SBR+ and antenna chamber measurements

Depending on the radar modulation parameters and the spatial points that the conic reflector traverse in the field of view, SBR+ simulation environment generates the transient files. Therefore, the number of transient files generated depends on the number of spatial points that the conic reflector is rotated. These transient files are fed to the python script that generates the rawData in the required data structure (Nr.Samples x Nr.Chirps x Nr. Rx channels x Nr. Spatial points) that can be processed in MATLAB.

A second rawData set is acquired in the anechoic chamber using the same radar modulation parameters. Both rawData sets (SBR+ simulation and chamber measurement) are imported into MATLAB, where the target parameters (range, Doppler, phase, angle) are extracted and an independent calibration matrix is generated for each environment. After calibration, the radar plots (magnitude, phase, angle estimation, angular error) obtained from simulation and measurement are compared. A typical calibration matrix produced by MATLAB is given in Equation 19.

$$H = \begin{bmatrix} -0.0717 - 0.3846i & 3.13e^{-17} + 0i & -0.0717 - 0.3846i \\ -0.0169 + 0.0389i & 1 + 0i & -0.0169 + 0.0389i \end{bmatrix}$$

Equation 19 Calibration matrix

10.1.3.1 Algorithm flow diagram for field of view (FoV) angle plots

From the previous section, rawData generated from the SBR+ simulation platform and anechoic chamber is given to the MATLAB environment for further post-processing. The calibration script needs to be updated based on the radar parameters (Nr. Samples, Nr. Chirps, sampling rate, time delay between chirps, and so on) and the antenna parameters (Nr. Spatial points, angle sweeps). Radar chipset variant ("BGT60CUTR13AIP") needs to be mentioned, so that the relevant Tx and Rx positions are taken into consideration for generating the synthetic signal.

After modifying the radar and antenna parameters, rawData is fed to range FFT and Doppler FFT blocks so that the target is localized across appropriate range and Doppler bins. For reference, zero Doppler axis in the range Doppler map is plotted. Axis scaling was taken care of by the helper functions. Target peak at the desired range bin is observed and cross verified. This specific target range bin index is selected so that target magnitude and phase is observed at the peak. Based on the target range, Tx and Rx antenna positions, chipset orientation, and angle of rotation, synthetic target phases were generated from the model based on different spatial points.

Now, the target phases coming from SBR+ simulation environment are compared against the synthetic phases from the model for different spatial points and fed to the minimum-norm least-square solution (lsqminnorm MATLAB function) to generate the calibration matrix based on SBR+ simulation rawData. This calibration matrix is applied to data from SBR+ platform to generate the FoV angle plot.

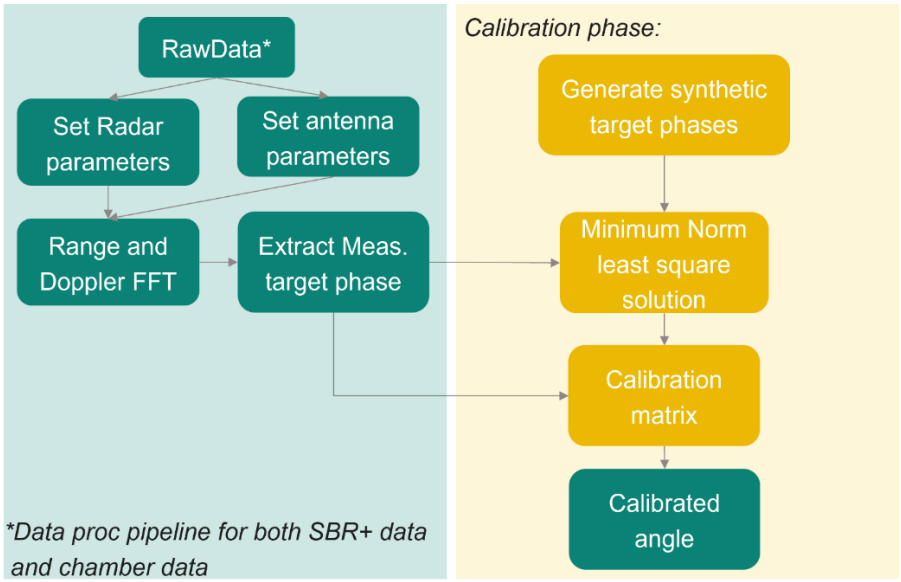


Figure 132 Top-level data pipeline view based on data from ANSYS SBR+ and antenna chamber measurements

The same procedure is applied to the chamber rawData to generate a chamber-specific calibration matrix and the corresponding FoV angle plot. Figure 132 shows the top-level pipeline for both environments. Figure 133 and Figure 134 compare the FoV angle and the angular error obtained from SBR+ simulation and from anechoic-chamber measurement.

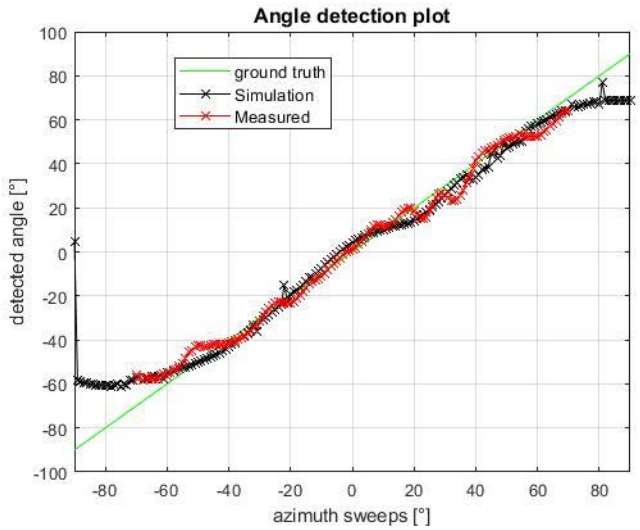


Figure 133 E-plane field of view (FoV) angle comparison plot between SBR+ simulation and anechoic chamber measurement with V3 RF shield board

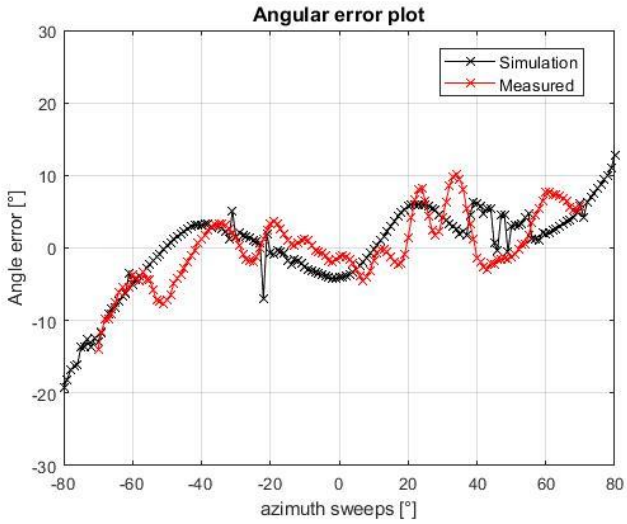


Figure 134 E-plane angular error comparison plot between SBR+ simulation and anechoic chamber measurement with V3 RF shield board

Within the calibrated bandwidth, the SBR+ simulation reproduces the AoA and angular-error trends measured in the anechoic chamber after tape-out. The same simulation framework can be reused to evaluate the impact of a customer-specific radome on the angular response and to support radome material and geometry selection.

10.2 Radar shield PCB setup

As discussed at the beginning of this section, the space allocated to the radar PCB is a key factor influencing radar performance. The PCB dimensions should therefore be assessed relative to the free-space wavelength at 60 GHz:

$$\lambda_0 = \frac{c}{f} \approx 5 \text{ mm}$$

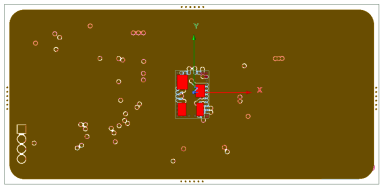
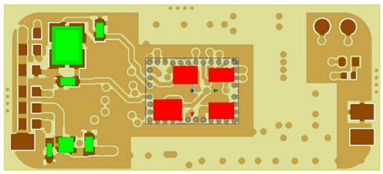
Equation 20 Free-space wavelength calculation

Based on their electrical size, PCBs can be classified into two categories:

- **Small PCBs:** Dimensions **below** five free-space wavelengths, corresponding to approximately 25 mm
- **Large PCBs:** Dimensions **above** five free-space wavelengths, corresponding to approximately 25 mm

The boundary between the two cases is set at $5\lambda_0 \approx 25$ mm, the dimension above which the surface-wave and parasitic-reflection contributions from the shield top metal become negligible relative to the direct radiated path. Table 38 summarizes the topology requirements of each case.

Table 38 PCB topology requirements based on PCB size relative to the free-space wavelength at 60 GHz

Large PCB (largest side ≈ $10\lambda_0$ ≈ 47 mm)	Small PCB (largest side ≈ 3–$5\lambda_0$ ≈ 15 mm to 25 mm)
22.9 mm × 47.9 mm	10.8 mm × 24.5 mm (RF shield V3)
	
Top metal: fan-in pads and full ground plane	Top metal: fan-in pads and ground with cutout around sensor
Stack-up: immune to prepreg thickness and Dk variations	Stack-up: top prepreg (PPG) ≈ 75 µm to 85 µm Top prepreg dielectric constant Dk = 4.4 or higher
Drills: PTH around the sensor	Drills: PTH around the sensor
Parasitic reflections are less critical. A continuous top ground plane is sufficient	Cutout in the top metal and thin prepreg are essential to suppress parasitic reflections

10.3 Small PCB – Largest side ≤ $5\lambda_0$ (≈25 mm)

When the largest PCB dimension is equal to or below $5\,\lambda_0$ (≤ 25 mm at 60 GHz), parasitic reflections between the target and the top metal of the shield concentrate near the sensor and degrade the angular response.

Two design measures are required to compensate for this effect and still achieve a wide FoV response:

- A thin top prepreg to reduce the excitations of substrate modes
- A cutout in the top ground layer around the radar chip to reduce the unwanted back and forth reflections

These measures are applied in the Infineon evaluation kit, which consists of a radar baseboard into which the RF shield board is plugged, as shown in Figure 135.

RF shield V3 (10.8 mm × 24.5 mm, largest side ≈24.5 mm ≈ $5\,\lambda_0$) is the design example used in the following subsections. In the following subsections, the corresponding E- and H-plane responses are presented.

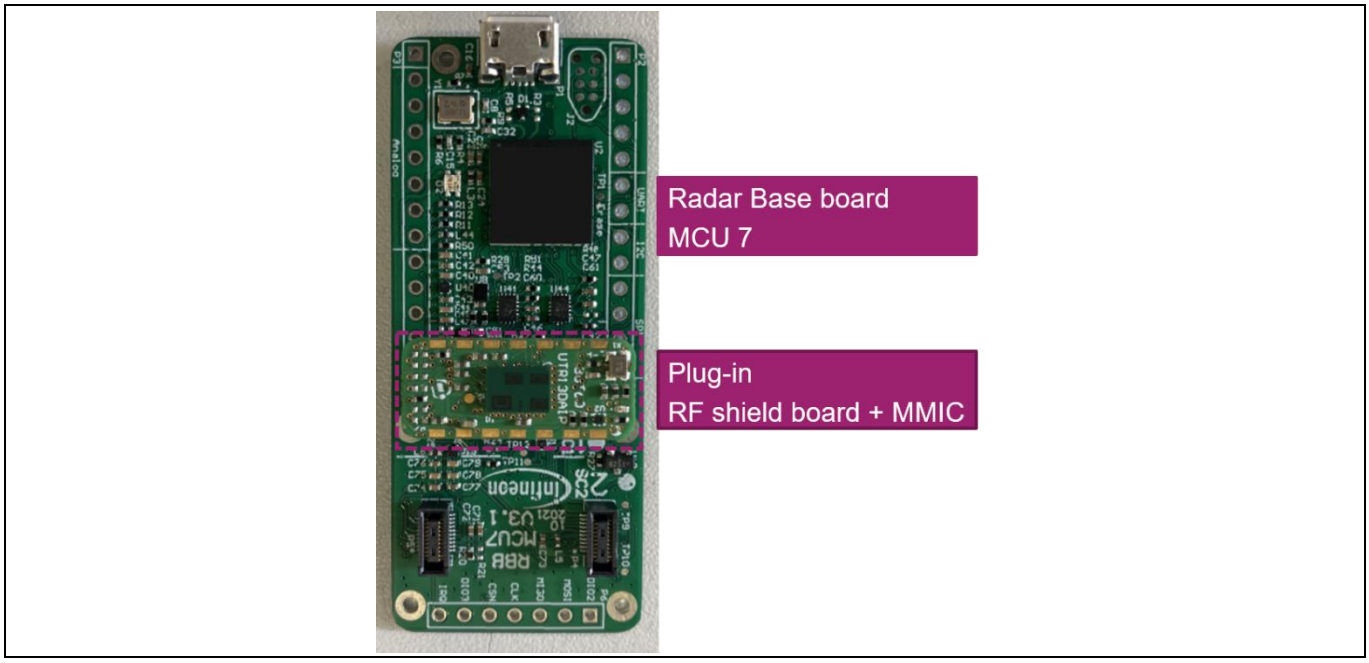


Figure 135 Infineon evaluation module including radar baseboard MCU 7 plus and RF shield PCB

10.3.1 Radar performance in E-plane

The E-plane configuration consists of TX as the transmitter and RX1/RX3 as the two receivers, separated by half a wavelength, as defined in Figure 136. The E-plane provides the widest field of view of the XENSIV™ BGT60CUTR13AIP in presence-detection mode. Table 39 shows the simulated and measured AoA and the angular error after mutual-coupling calibration versus theta. Table 40 presents the magnitude of the composite TX/RX signal for the same sweep. Since the Angle of Arrival (AoA) is reconstructed from the phase difference between TX/RX1 and TX/RX3, a linear phase difference versus theta is the prerequisite for a linear phase difference and a minimum angular error. Moreover, the TX/RX magnitude is recommended to remain 15 dB below its broadside value; a deeper drop degrades the SNR and the range resolution at the affected angles. RF shield V3 achieves a FoV of ±70° with a maximum angular error of ±10° (Table 39) and a normalized TX/RX magnitude better than –10 dB across the same FoV (Table 40).

In conclusion, the design guidelines of a small RF shield PCB whose size is less than 25 mm are summarized in the following design steps:

- Choose a PCB stack-up with a thin top prepreg to reduce the substrate modes
- If needed, consider adding cutouts in the layout of the top metal to reduce the back-and-forth reflections in radar mode. Otherwise, use a complete top ground layer with shielding drills connecting all the metal layers on the PCB
- Reduce the components close to the radar chip
- Perform radar mode simulations to double check the linearity of the phase response

These design steps are detailed in the following subsections:

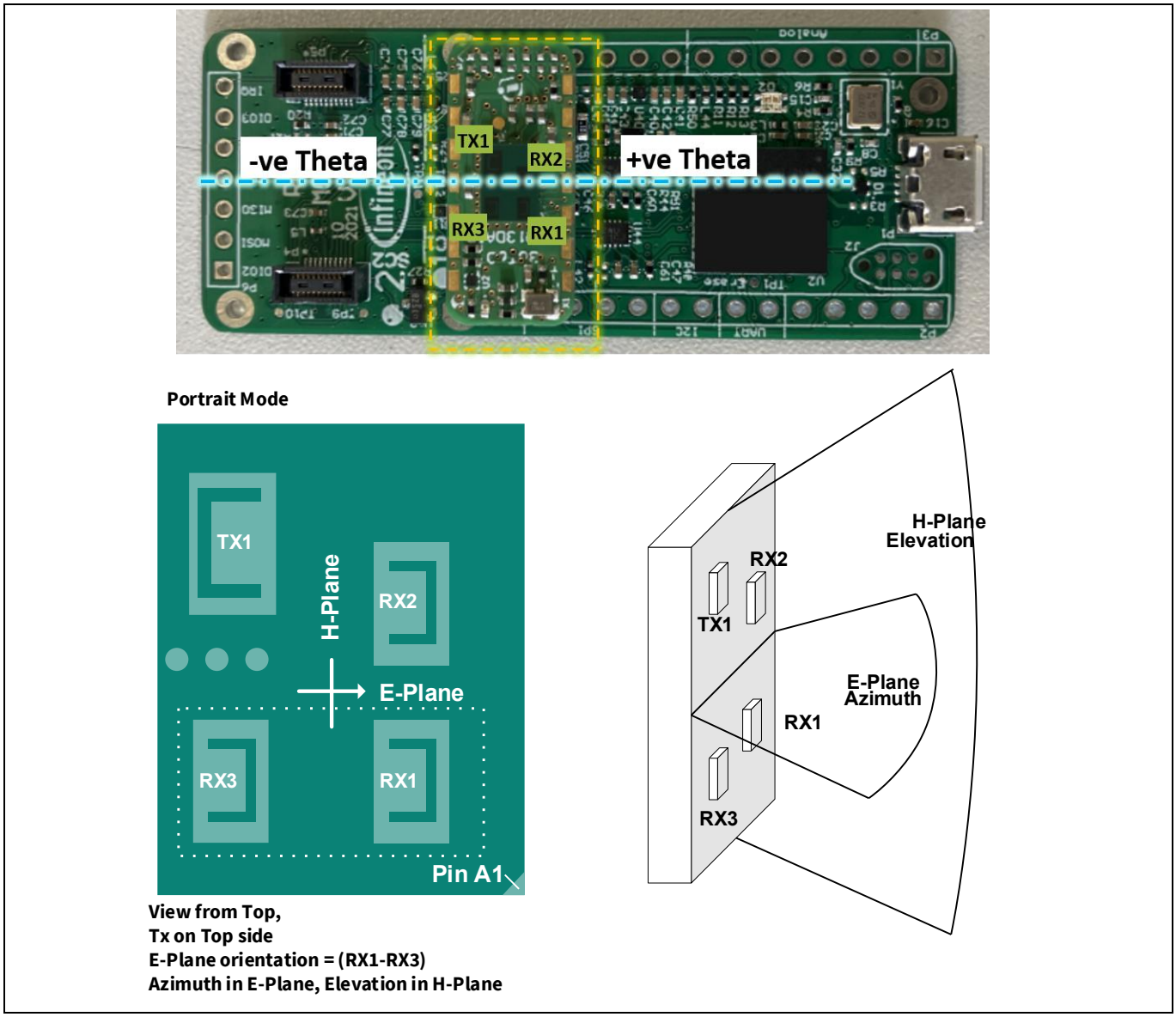


Figure 136 E-plane - definition of shield orientation, antenna position, positive and negative theta angles, USB connector on right side, MMIC – TX on top-left side

10.3.1.1 Step 1 – Choose the right PCB stack up

The shield-PCB stack-up directly affects the linearity of the AoA versus azimuth. A thick top prepreg supports parasitic substrate modes between the top ground and the buried reference plane; these modes couple to the antenna and distort the radiated phase front. Reducing the top prepreg to ≤ 75 µm raises the cut-off of the dominant parasitic mode well above the operating band and restores a linear AoA response, as implemented in RF shield V3.

10.3.1.2 Step 2 – Optimize the layout of the top metal layer

Even with an optimized stack-up, a continuous top ground extending along the E-plane axis of a compact shield produces an angular error of up to 15°. The dominant mechanism is the coherent back-and-forth reflection between the target and the top-metal/MCU-board surfaces, which adds a parasitic component to the received signal and distorts the phase difference between TX/RX1 and TX/RX3 in the E-plane.

The best practice is to reduce the effective reflective surface along the E-plane by making a metal cutout in the top metal around the radar chip. RF shield V3 implements such a cutout; both SBR+ simulation and anechoic-chamber measurement confirm a linear AoA response and a maximum angular error of ±10° across a field of view of ±70°. It is important to note that there is a drawback of using cutouts in radar PCB. Firstly, the cutout exposes the first prepreg layer, this results in a dependence on the dielectric constant of the prepreg material. The best performance is achieved when the dielectric constant of the first prepreg is at least 4.4 like FR4 epoxy. If several PCB suppliers are considered for the radar PCB, the stack up parameters should remain unchanged across suppliers.

Table 39 E-plane angle of arrival and angular error: simulations vs. measurements with RF shield V3

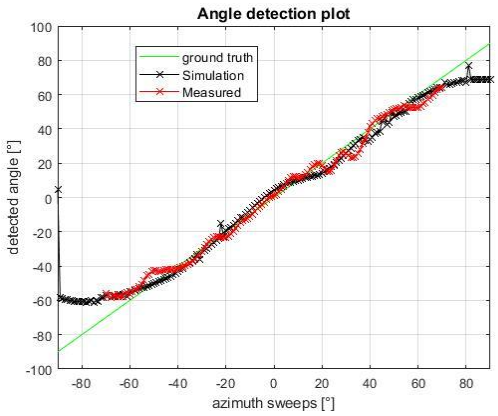
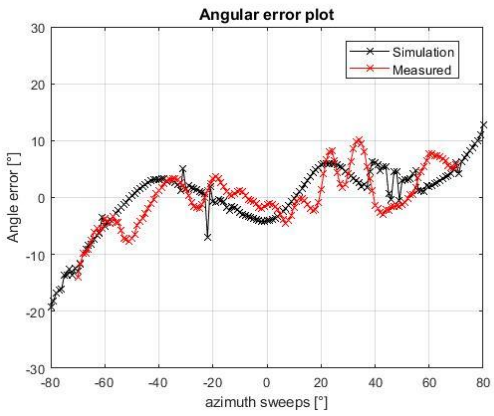
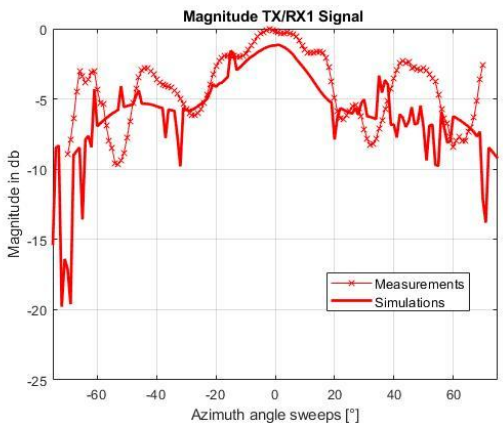
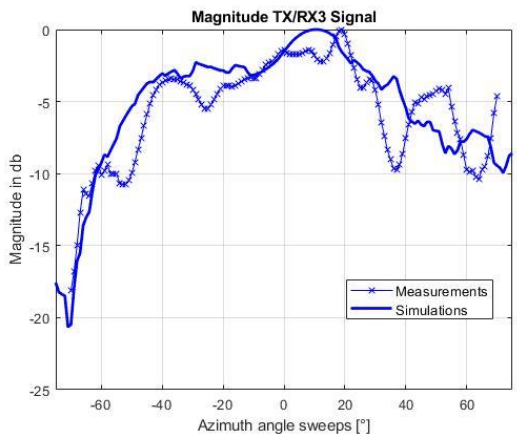
RF shield Version	Angle of arrival (AoA)	Angular error using mutual coupling calibration
Shield V3		

Table 40 E-plane signal magnitude: simulations vs. measurements with RF shield V3

RF shield Version	Magnitude of TX/RX1 signal in E-plane	Magnitude of TX/RX3 signal in E-plane
Shield V3		

10.3.2 H-plane performance – Small PCB

The H-plane configuration consists of TX as a single transmitter and RX1/RX2 as the two receivers, horizontally separated by half a wavelength, as defined in Figure 137. The H-plane FoV is intrinsically narrower than the E-plane FoV – a known property of the microstrip-patch antenna used in the XENSIV™ BGT60CUTR13AIP sensor. Table 41 lists the simulated and measured AoA and angular error; Table 42 shows the magnitude of the composite TX/RX signal versus theta. As in the E-plane, AoA linearity is governed by the linearity of the difference between the phase of TX/RX1 composite signal and the phase of the TX/RX2 composite signal. The drop in the composite signal magnitude from the maximum should remain less than 15 dB in order to avoid degradation in the signal to noise ratio (SNR).

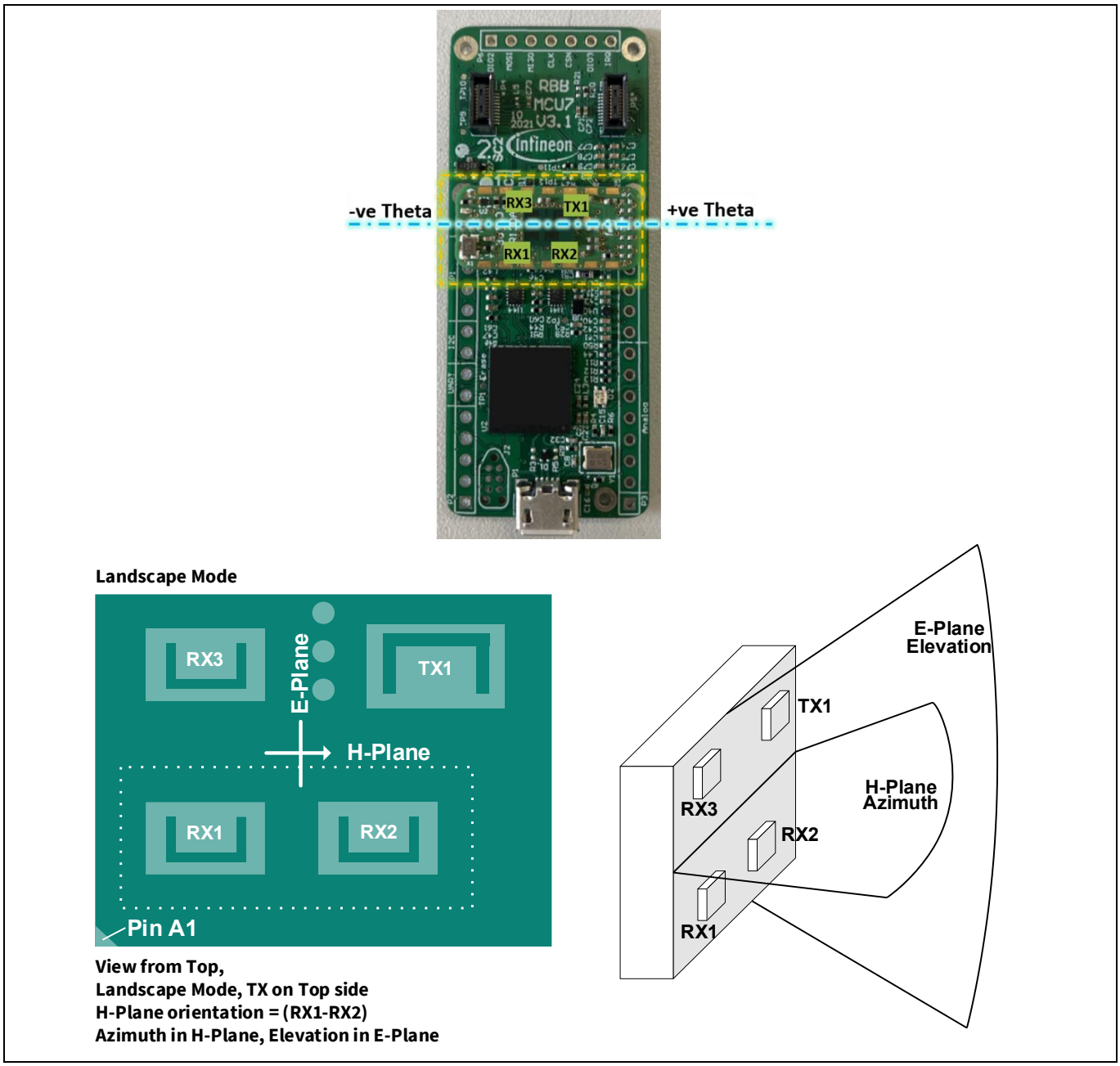


Figure 137 H-plane - definition of shield orientation, antenna position, positive and negative angles, USB connector on bottom side, MMIC – TX on top-right side

Table 41 H-plane angle of arrival and angular error: simulations vs. measurements with RF shield V3

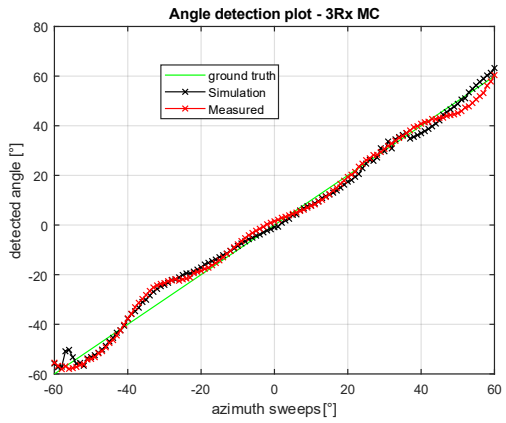
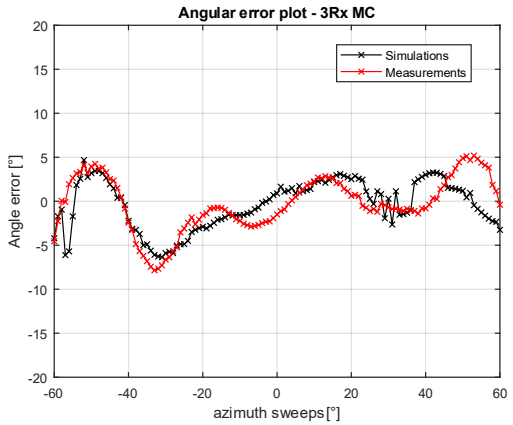
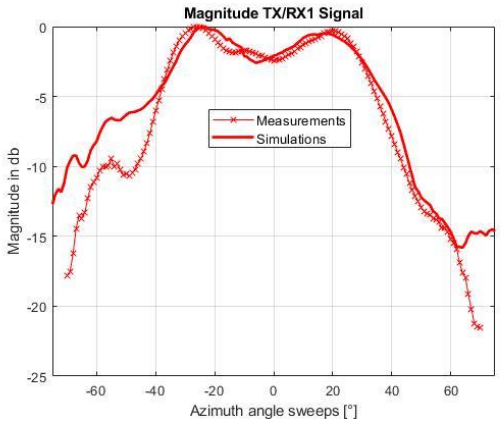
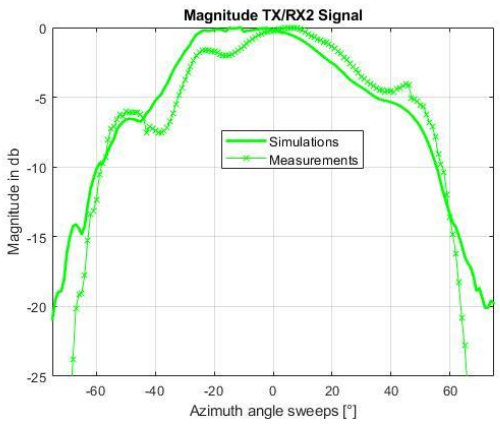
RF shield Version	Angle of arrival (AoA)	Angular error using mutual coupling calibration
Shield V3		

Table 42 H-plane magnitude of TX/RX signal: Simulations vs. measurements with RF shield V3

RF shield Version	Magnitude of TX/RX1 signal in H-plane	Magnitude of TX/RX2 signal in H-plane
Shield V3		

10.4 Large PCB – Largest side > 5λ_0 (≈25 mm)

When the largest PCB dimension exceeds 5 λ_0 (> 25 mm at 60 GHz) the back-and-forth reflections between the target and the radar surroundings are spread over a relatively large reflective surface, so the parasitic contributions at the sensor location are weaker than in the compact small PCB case. A continuous top ground plane with fan-in pads is therefore sufficient, and the angular response is largely insensitive to moderate prepreg thickness variations.

Figure 138 shows the layout of the top metal where the sensor is centered in the middle of the PCB. Figure 139 shows a closer view of the fan-in pads. These pad lines connect the PCB feedlines with the sensor solder balls. Since these connections are ohmic and the signals are non-RF signals, the pad lines should be as short as possible and preferably routed underneath the package in a fan-in approach, as shown in Figure 139. This topology helps suppress unwanted parasitic modes.

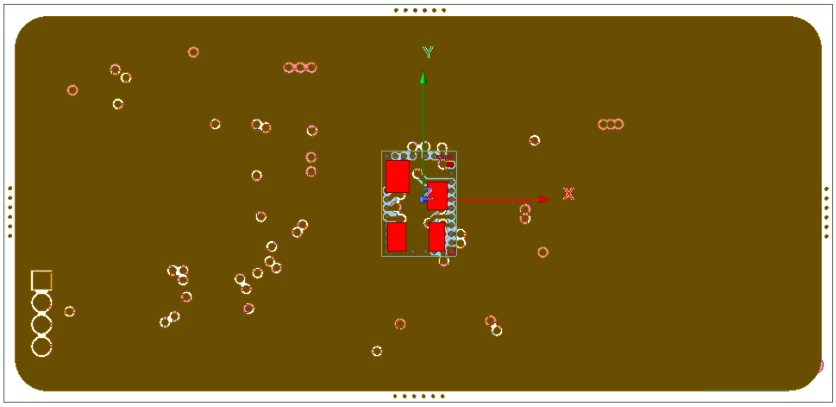


Figure 138 Large PCB (22.9 mm × 47.9 mm) radar shield board

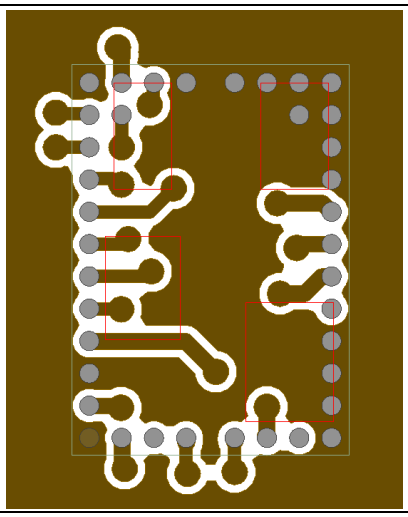


Figure 139 Example of fan-in pads on a large PCB

10.4.1 E-plane performance – Large PCB

Figure 140 shows, from left to right, Simulated TX/RX signal magnitude (left), Measured TX/RX signal magnitude (center), angular error (right) versus azimuth Theta angles. The angular error remains within ±10°, and the normalized magnitude is within –15 dB across a field of view from -70° to +80°. This topology is recommended when a wide field of view is required.

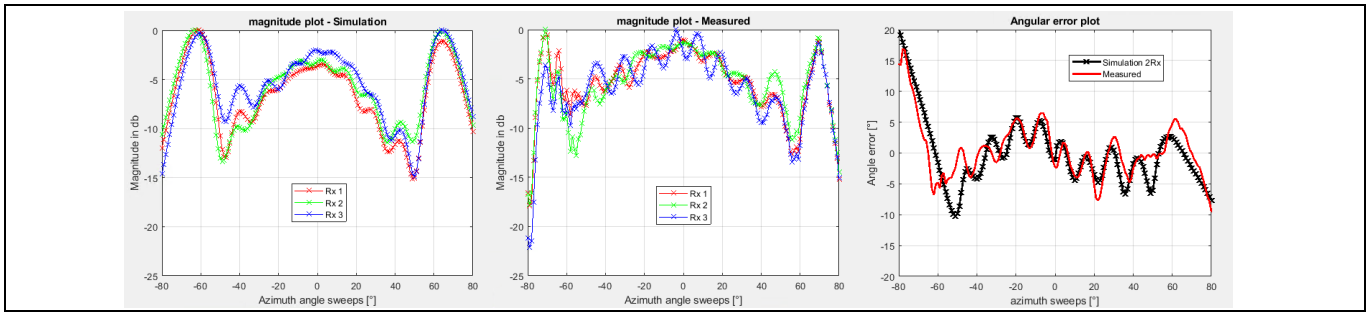


Figure 140 E-plane radar performance for large PCB: Simulated TX/RX signal magnitude (left), Measured TX/RX signal magnitude (center), angular error (right)

10.4.2 H-plane performance – Large PCB

Figure 141 shows the corresponding H-plane response: Simulated TX/RX signal magnitude (left), Measured TX/RX signal magnitude (center), angular error (right) versus azimuth Theta angles. The H-plane field of view is intrinsically narrower than the E-plane FoV for the microstrip-patch. From -60° to +55°, the magnitude plot remains symmetrical above -15 dB and the angular error stays within the ±10° specification.

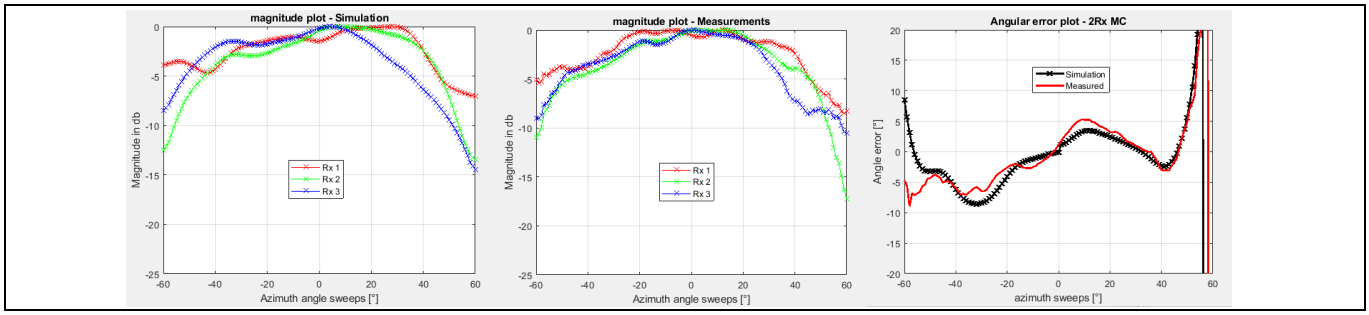


Figure 141 H-plane radar performance for large PCB Simulated TX/RX signal magnitude (left), Measured TX/RX signal magnitude (center), angular error (right)

10.5 Impact of radome on radar performance

10.5.1 Introduction

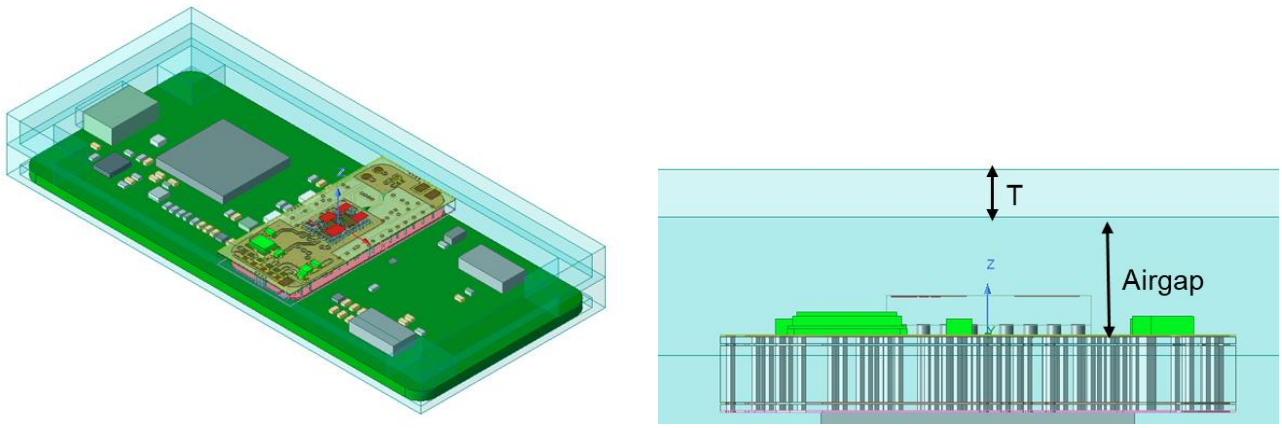
The product enclosure includes a plastic cover (radome) of defined shape, thickness and dielectric properties that encloses the radar PCB and modifies the angular response. The following section provides design recommendations for the radome geometry.

10.5.2 SBR+ simulation setup

The optimized RF shield V3, introduced in the previous sections as an example of a small PCB, is reused here to evaluate the impact of a plastic radome on radar performance. A flat plastic radome with thickness T is positioned above the PCB, with an airgap between its inner surface and the top surface of the PCB, as illustrated in Figure 142. Figure 142(a) shows the complete radome assembly, including the MCU board, while Figure 142(b) defines the radome thickness and airgap.

Radome thickness and airgap are two key design parameters that influence radar performance. The radome's dielectric constant (Dk) and curvature also play important roles at 60 GHz. This section examines the effects of thickness and airgap for a flat radome with Dk = 2.7 and a dissipation factor of Df = 0.007.

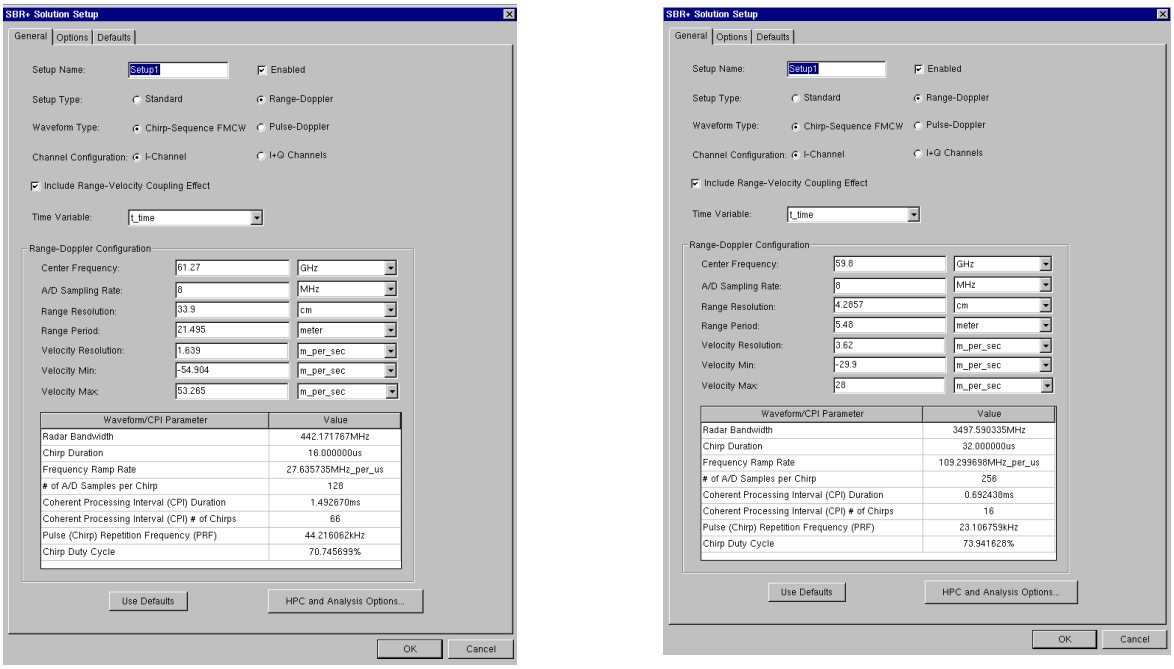
The conclusions presented in this section are specific to flat-profile radome with similarly low dielectric constants and comparable dielectric losses (Dk ≈ 2.7, Df ≈ 0.007) and should not be generalized to other material properties or radome geometries.



(a) (b)

Figure 142 A radome assembly including the MCU board, shield PCB, and radar sensor, along with the radome thickness (T) and the distance (Airgap) between the shield PCB and the bottom surface of the radome

The chirp-sequence FMCW I-channel radar operation is simulated for the two standard bandwidths, 500 MHz and 3500 MHz. The corresponding SBR+ solution setups are shown in Figure 143 (a) and (b), respectively. The target is a conic reflector with a –7 dBsm radar cross-section (RCS).



(a) 500 MHz bandwidth (b) 3500 MHz bandwidth

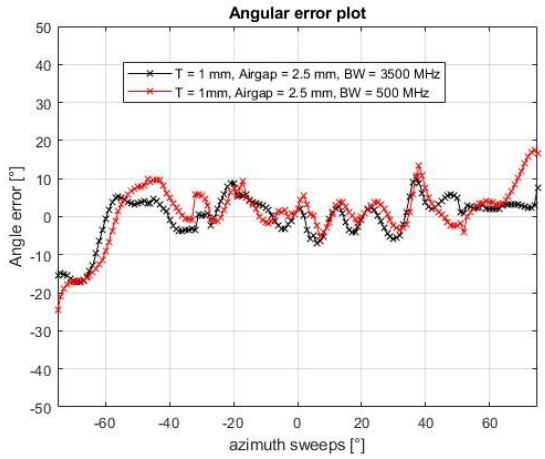
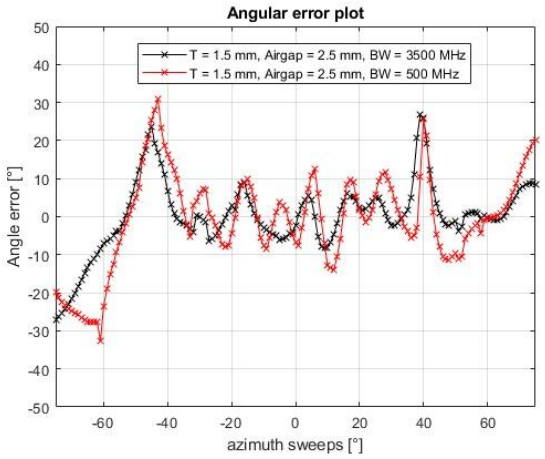
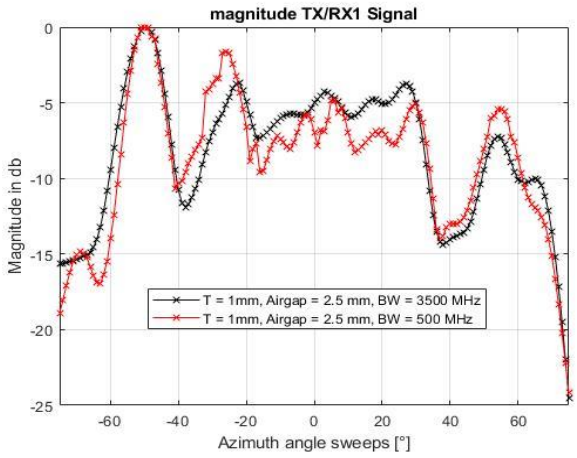
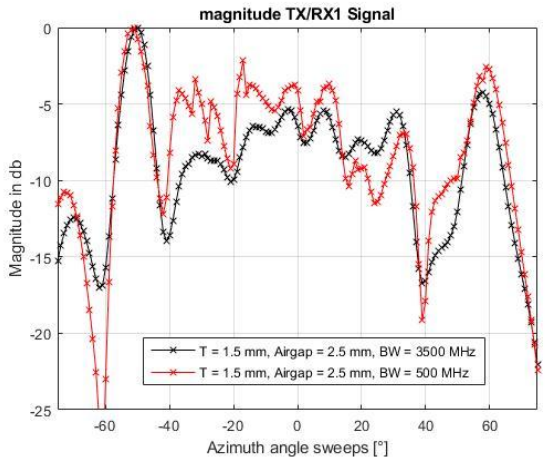
Figure 143 SBR+ solution setup for chirp-sequence FMCW I-channel radar at 500 MHz and 3500 MHz bandwidth

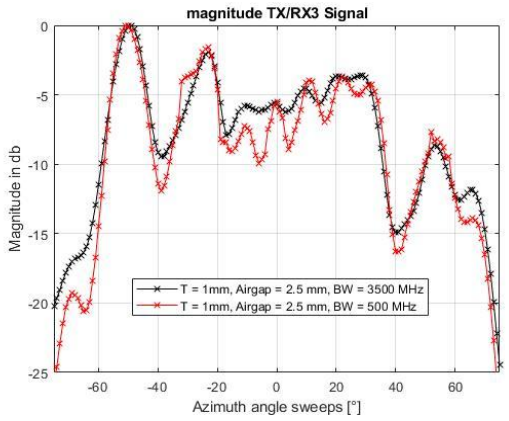
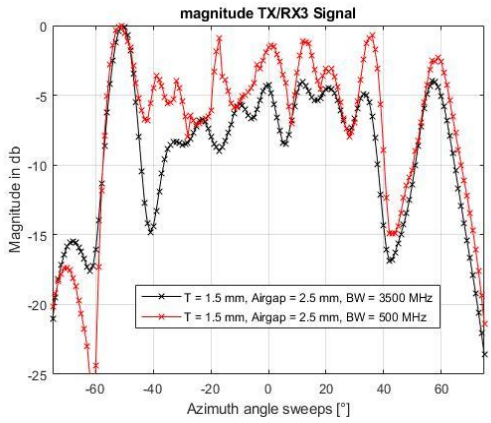
10.5.3 Recommended radome thickness

Radar-mode simulations are performed for two radome thicknesses, T = 1 mm and T = 1.5 mm, which represent the most common values offered by plastic-cover suppliers. The airgap is fixed at 2.5 mm (= $\lambda_0/2$ at 60 GHz). The radome material has a dielectric constant $\varepsilon_r = 2.7$ and a loss tangent $\tan\delta = 0.007$. Both 500 MHz and 3500 MHz radar bandwidths are simulated. Table 43 reports the simulated angular error and the magnitude of the scattered signals (TX/RX1 and TX/RX3). With T = 1 mm, the angular error stays within ±10° from –60° to +70°. With T = 1.5 mm the response is reliable only between –40° and +40°; beyond these angles, the angular error increases and a simultaneous drop in the TX/RX1 and TX/RX3 magnitude indicates that both amplitude and phase deteriorate. A thick radome is therefore unsuitable in this CAD setup when a wide FoV is required.

For a flat radome with Dk = 2.7, Df = 0.007, the cover thickness of 1 mm is recommended as a starting point for 60 GHz operation; the response is then stable for both the 500 MHz and the 3500 MHz bandwidth.

Table 43 E-plane angular error and signal magnitude plots with radome thickness (T) 1 mm, 1.5 mm and 2.5 mm airgap at 500 MHz and 3500 MHz bandwidths

Recommended Setup: T = 1 mm Airgap = $\lambda_o/2$ (2.5 mm at 60 GHz)	**T = 1.5 mm Airgap = $\lambda_o/2$ (2.5 mm at 60 GHz)**
	
	

Recommended Setup: T = 1 mm Airgap = $\lambda_o/2$ (2.5 mm at 60 GHz)	T = 1.5 mm Airgap = $\lambda_o/2$ (2.5 mm at 60 GHz)
	

10.5.4 Recommended airgap

With the radome thickness fixed at 1 mm, the airgap is varied to evaluate its impact on radar performance. The reference airgap of 2.5 mm, corresponding to half the free-space wavelength at 60 GHz (left column of Table 43), is compared with quarter-wavelength (1.25 mm) and three-quarter-wavelength (3.75 mm) airgaps in Table 44. The 3.75 mm airgap (right column of Table 44) produces the poorest radar response, with more pronounced effects from multiple reflections involving the sensor surroundings. Consequently, the usable FoV is limited to ±20°. The magnitude drop at large off-axis angles is deeper with the 500 MHz bandwidth than with the 3500 MHz bandwidth. Based on these results, an airgap larger than half the free-space wavelength are strongly not recommended for the configuration investigated.

The smaller airgap of 1.25 mm (left column of Table 44) provides a relatively stable response magnitude across the full FoV for both bandwidths. However, the half-wavelength airgap of 2.5 mm remains the preferred compromise in terms of angular error. The quarter-wavelength configuration is acceptable only when product-size constraints prevent the use of a half-wavelength airgap, as the angular error can reach ±20° at the edges of the FoV.

10.5.5 Conclusion

As shown in Figure 144, adding a radome degrades the angular accuracy when compared with the standalone PCB. As a rule of thumb, an airgap of approximately half the free-space wavelength – 2.5 mm at 60 GHz is recommended between the top surface of the PCB and the bottom surface of the radome. For the configurations investigated, this spacing generally keeps the angular error within ±10° and helps maintain the normalized TX/RX magnitude above the −15 dB threshold across the FoV. Nevertheless, deviations may still occur at the far angles, as illustrated by the red curve in Figure 144.

The achievable radar performance depends on the complete product geometry and the radome's electrical properties at 60 GHz. Radar-mode simulations incorporating the full product CAD model and PCB are therefore strongly recommended as part of the hardware design process to assess the radar response and identify potential performance limitations.

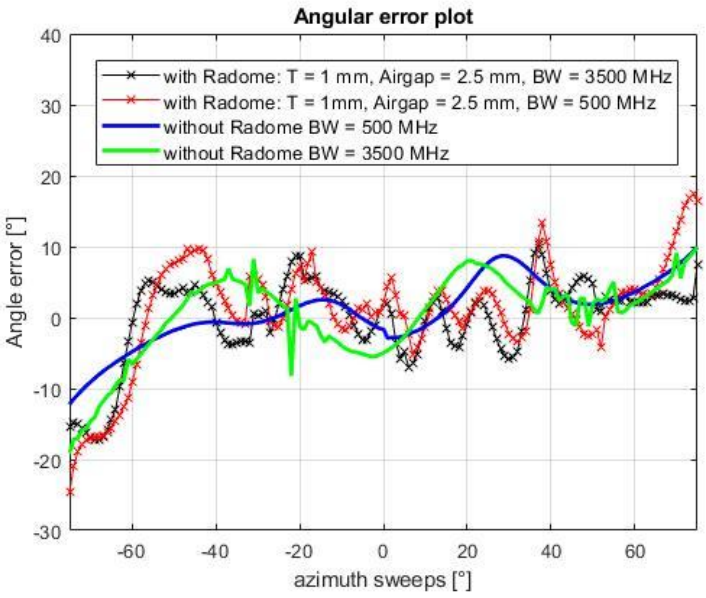
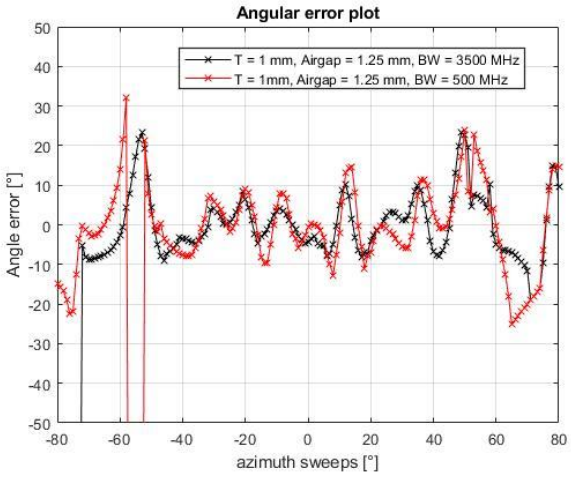
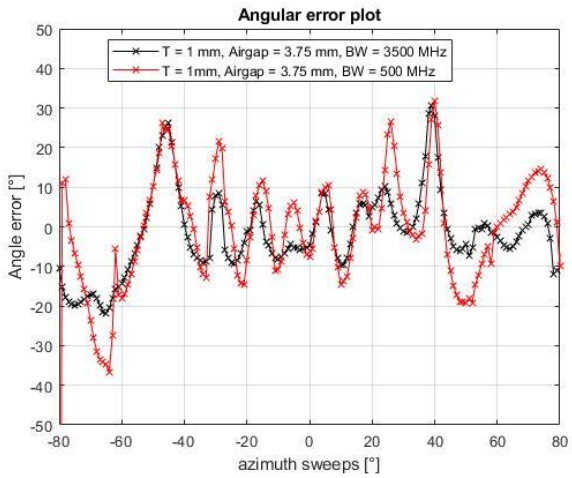
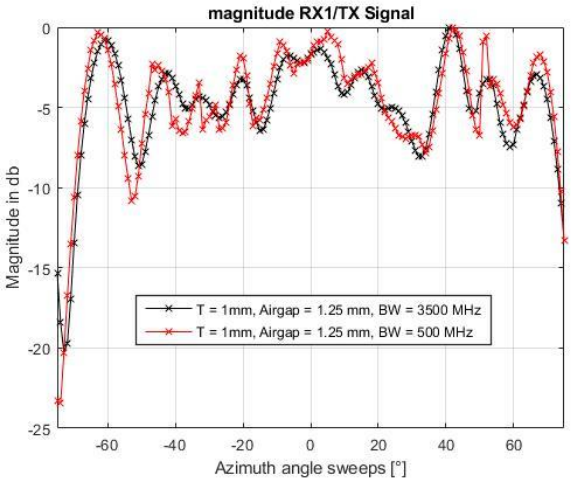
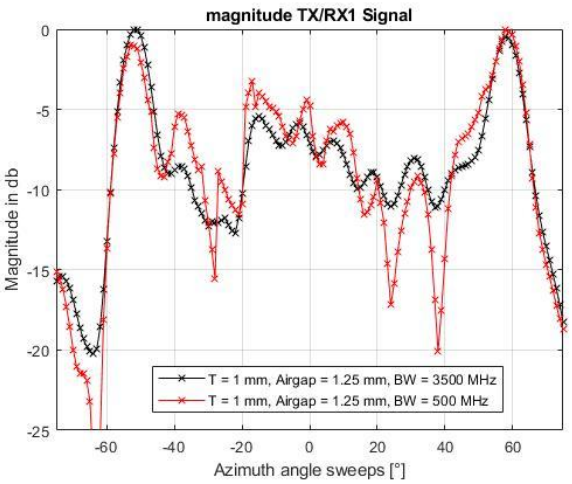
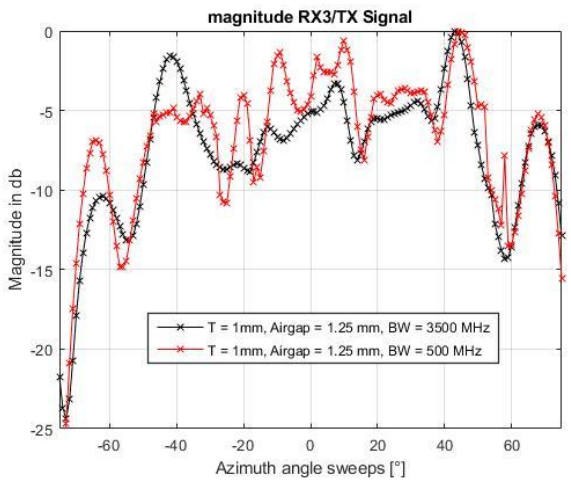
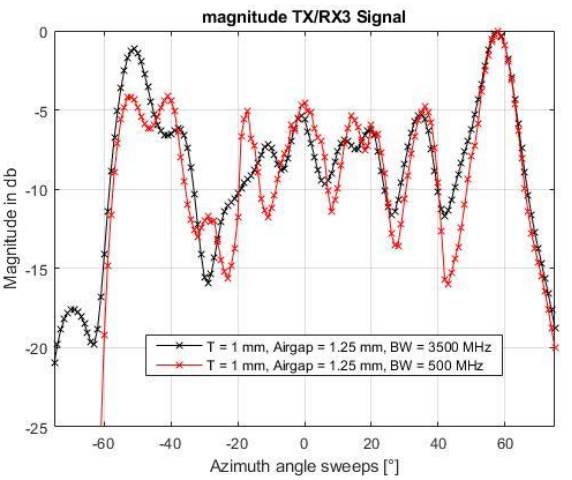


Figure 144 E-plane angular error with the V3 shield board with and without radome at 500 MHz and 3500 MHz bandwidth operation

Table 44 E-plane angular error and signal magnitude plots with radome thickness (T) and airgap $\lambda_o/4$ (1.25 mm at 60 GHz), 3 $\lambda_o/4$ (3.75 mm at 60 GHz) at 500 MHz and 3500 MHz bandwidths

Airgap = $\lambda_o/4$ (1.25 mm at 60 GHz) T = 1 mm	Airgap = 3 $\lambda_o/4$ (3.75 mm at 60 GHz) T = 1 mm
	
	
	

10.6 Design guidelines of CAD and PCB hardware integration

This final section summarizes general design guidelines for integrating a 60 GHz radar sensor into a product assembly. For a PCB-mounted sensor covered by a plastic radome, as illustrated in Figure 145, the integrated radar performance is primarily governed by three factors: the PCB top-layer metallization, the PCB-to-radome airgap, and the radome's dielectric properties and geometry, including its dielectric constant (Dk), dissipation factor (Df), and curvature. These factors must be considered jointly to achieve performance comparable to that of the evaluation kit (EVK) boards.

For operation at 60 GHz, the nominal airgap, defined as the separation between the PCB top surface and the radome inner surface, should be approximately half the free-space wavelength, corresponding to 2.5 mm. A radome dielectric constant close to unity is desirable but generally not achievable with conventional solid materials. For materials such as Gorilla glass, Dk can reach approximately 7. Such high-permittivity materials can produce pronounced multiple reflections between the radome and the PCB top surface, reducing the usable field of view (FoV) and increasing angular estimation error. In such cases, the PCB top-layer metallization requires careful optimization to mitigate the resulting performance degradation.

Moreover, the PCB dimensions and layout topology significantly influence the achievable FoV and must therefore be considered as part of the overall radar integration design.

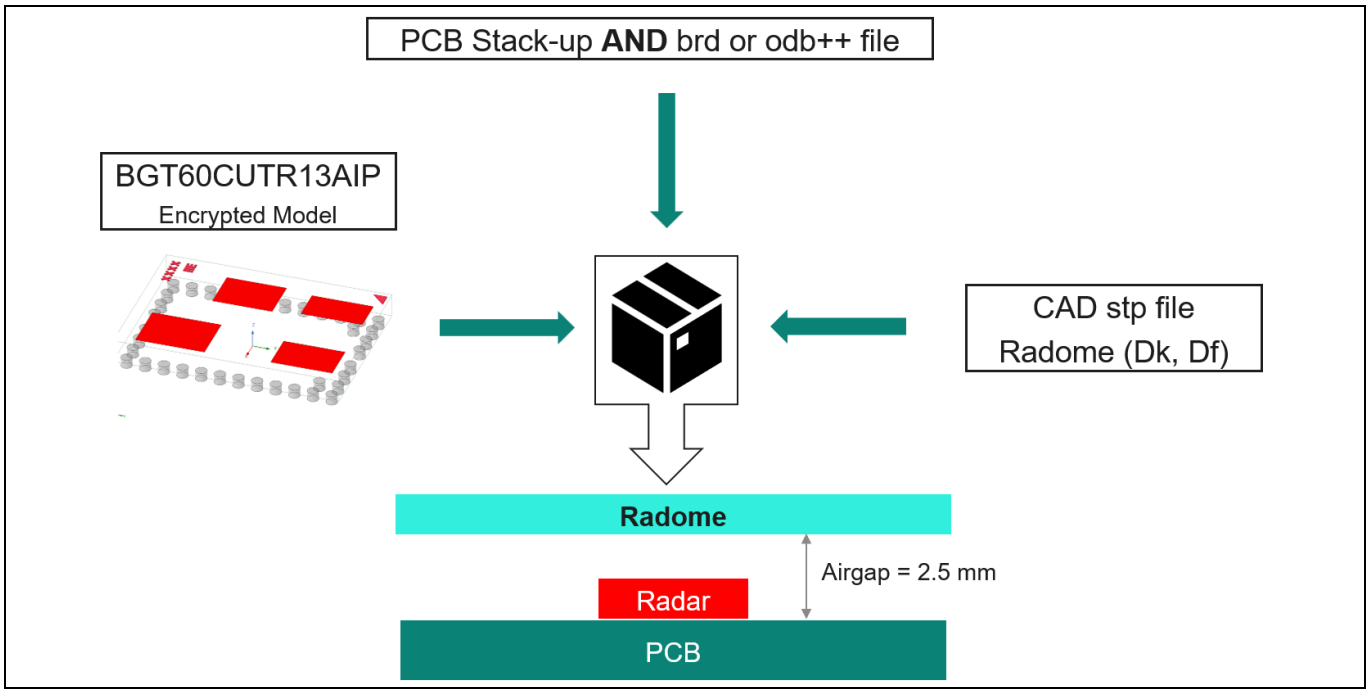


Figure 145 PCB and CAD integration flow with radar MMIC

The flowchart in Figure 146 summarizes the design decision process for radar integration based on the radome dielectric constant (Dk) and PCB size. The process begins by allocating the place holder for the sensor and PCB within the product CAD model, maintaining a nominal airgap of 2.5 mm between the PCB top surface and the radome inner surface. Wherever possible, the radar PCB should be positioned symmetrically relative to the radome to maximize the achievable azimuth field of view (FoV). An asymmetric placement is not recommended. The next step distinguishes between radome materials with Dk below 3 and those with Dk above 3. Within each category, the PCB is classified as small or large using the five-wavelength threshold. The small- and large-PCB configurations are summarized in Table 38 and discussed in detail in Sections Small PCB – Largest side ≤ $5\lambda_0$ (≈25 mm) and Large PCB – Largest side > $5\lambda_0$ (≈25 mm). The combination of radome Dk and PCB size determines the initial mechanical design to use in subsequent CAD optimization.

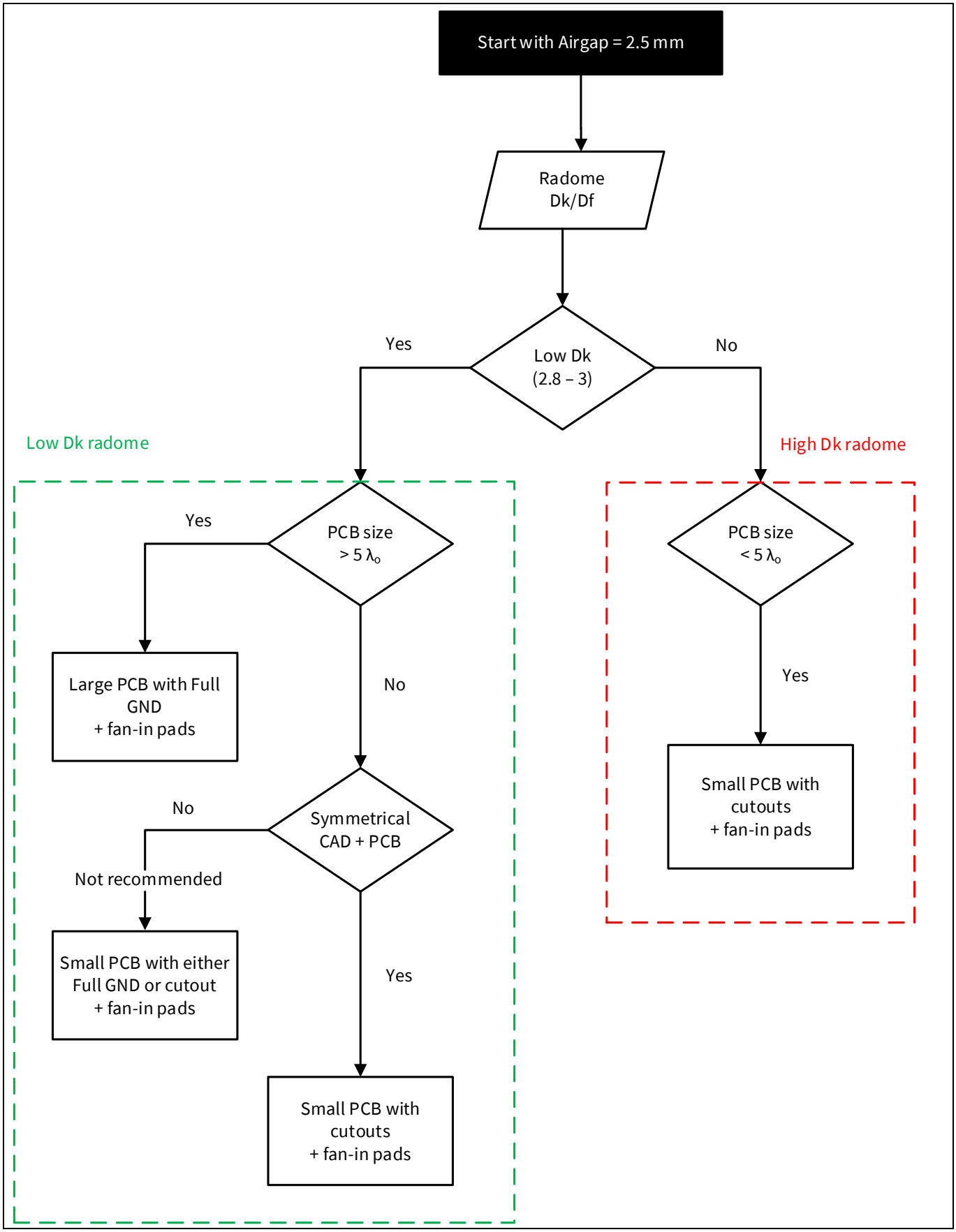


Figure 146 PCB and CAD integration flow diagram

10.7 Measurement setup

10.7.1 Introduction

The performance of antennas used in frequency modulated continuous wave (FMCW) radars designed with antenna in package (AiP) requires thorough characterization to ensure optimal system performance. This involves measuring the radiation pattern of both transmit and receive antennas to confirm that the actual radiation pattern matches the expected design.

Additionally, the effectiveness of the FMCW radar in detecting static targets must be validated. To achieve this, the radar system is tested to detect the same target at different angular locations with various feasible chirp bandwidths. This test determines the range of angles between which targets can be detected, also known as the field of view (FoV) of the radar system.

10.7.2 Measurement environments

Measurements can be performed in either an open-air measurement setup (OAMS) or an anechoic chamber. The latter is preferred for documenting and comparing the performances of various radar system designs due to its advantages, including:

- **Unaffected tests by electromagnetic waves and reflective surfaces:** The anechoic chamber provides a controlled environment that minimizes the impact of external electromagnetic waves and reflective surfaces, ensuring that the measurements are not influenced by external factors
- **Improved repeatability and efficiency of measurements:** The controlled environment and precise measurement setup of the anechoic chamber enable repeatable and efficient measurements, reducing the time and resources required for testing
- **Stable test environment:** The anechoic chamber provides a stable test environment that is not affected by changes in temperature, humidity, or other environmental factors, ensuring that the measurements are consistent and reliable
- **Easier standardization of the reference for performance of each device-under-test (DUT):** The anechoic chamber enables the standardization of the reference for each DUT performance, making it easier to compare the performance of different radar systems
- **Electromagnetic interference does not radiate outside:** The design of anechoic chamber ensures that electromagnetic interference generated during testing does not radiate outside the chamber, minimizing the risk of interference with other devices or systems
- **Tests with strong electric fields can be performed, if required:** The design of anechoic chamber allows for testing radar systems with strong electric fields, if required, without posing a risk to the surrounding environment

However, certain measurement test cases must be performed in OAMS to confirm the usability of the radar system as a suitable sensor for the desired application. This document focuses on measurements performed inside an anechoic chamber.

The following three sets of measurements are performed in the anechoic chamber:

- Tx characterization
- Rx characterization
- Radar mode measurement

Each measurement setup is elaborated in the following sections, including the measurement components, devices used, system parameters, and outcomes.

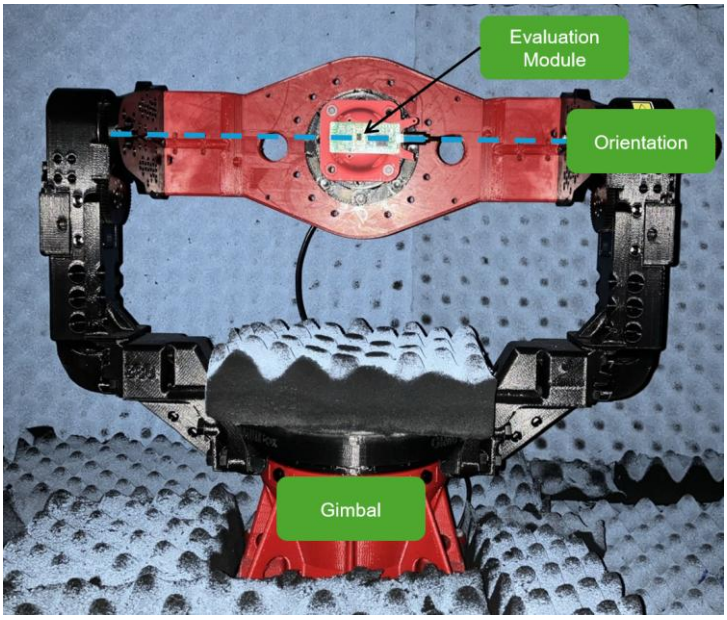


Figure 147 Sensor orientation during E-plane measurement

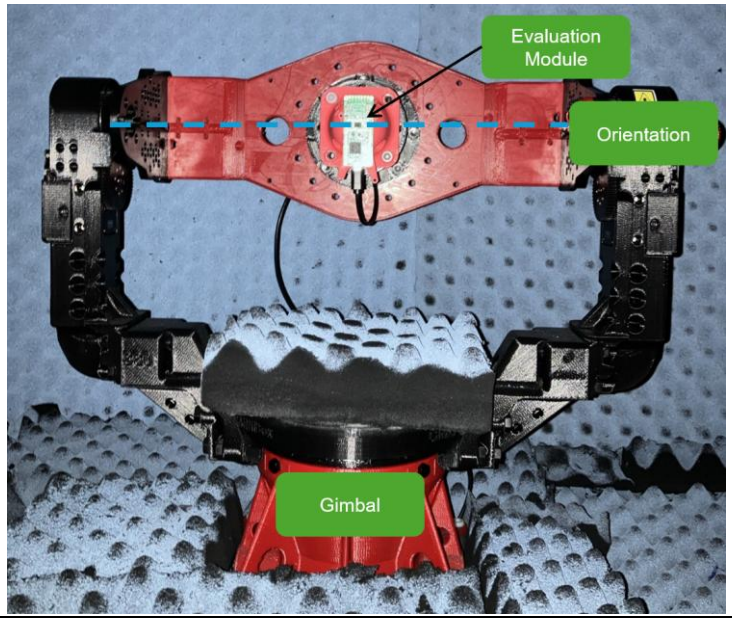


Figure 148 Sensor orientation during H-plane measurements

Each measurement on the XENSIV™ BGT60CUTR13AIP V3 shield is measured for both E- and H- cut-planes. The measurement setup and corresponding sensor orientations employed for E-plane and H-plane measurements are shown in Figure 147 and Figure 148, respectively. Furthermore, the shield orientation and antenna positions used during E-plane and H-plane measurements can be found in Figure 136 and Figure 137, respectively.

10.7.3 Tx characterization

The Tx characterization measurement aims to obtain the effective isotropic radiated power (EIRP) level and the gain pattern of the transmit antennas of the radar system. The measurement setup consists of the following devices and measurement components, as shown in Figure 149:

- Device under test (DUT)
- Gimbal
- Horn antenna
- Antenna adapter
- Connecting cable
- Smart mixer
- Spectrum analyzer

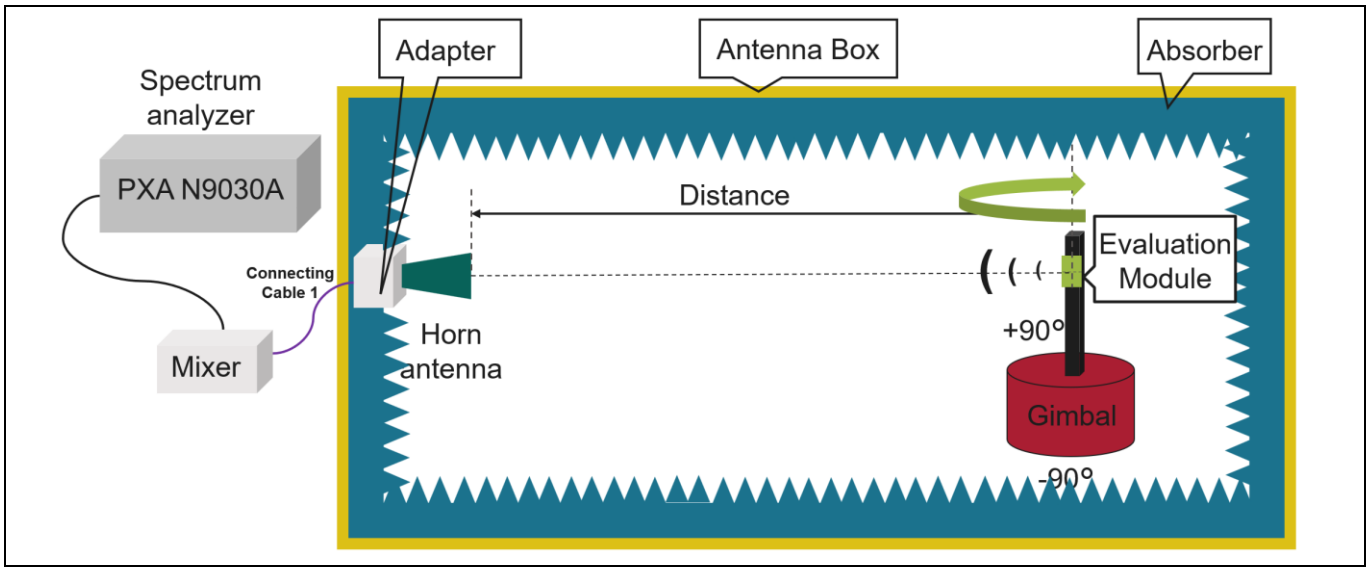


Figure 149 Measurement setup used to perform Tx characterization

The measurement process involves:

- **Setting the individual Tx antenna of the DUT to transmit at maximum power in continuous wave mode:** The Tx antenna is set to transmit at its maximum power level in continuous wave mode to ensure that the measurement is performed under optimal conditions. This is done to simulate the actual transmission scenario, where the radar system transmits a continuous wave signal
- **Using MATLAB scripts to control the spectrum analyzer, gimbal, and DUT to improve repeatability:** MATLAB scripts are used to control the Spectrum Analyzer, Gimbal, and DUT to ensure that the measurement is performed in a repeatable and efficient manner. This is done to minimize human error and ensure that the measurement is consistent
- **Measuring the distance between the horn antenna and the DUT to account for free-space path loss:** The distance between the horn antenna and the DUT is measured to account for free-space path loss, which affects the signal strength and accuracy of the measurement. Free-space path loss is a critical factor in radar systems, as it affects the signal strength and range of the system

- **Compensating for cable losses, adapter losses, and free-space distance losses to obtain the EIRP value:** The losses incurred due to the connecting cable, antenna adapter, and free-space distance are compensated to obtain the accurate EIRP value. This ensures that the measured EIRP value is accurate and representative of the radar system's performance

The Tx characterization measurement provides valuable information about the radar system's transmission capabilities, including the EIRP level, which is critical for determining the system's range and accuracy. De-embedding the output power from the EIRP level can also provide the expected gain pattern of the transmit antennas.

10.7.3.1 Measurement results

EIRP plot from the Tx characterization measurement for XENSIV™ BGT60CUTR13AIP V3 shield, performed at CW frequencies from 57 GHz to 63 GHz in step of 1 GHz, are shown in Figure 150.

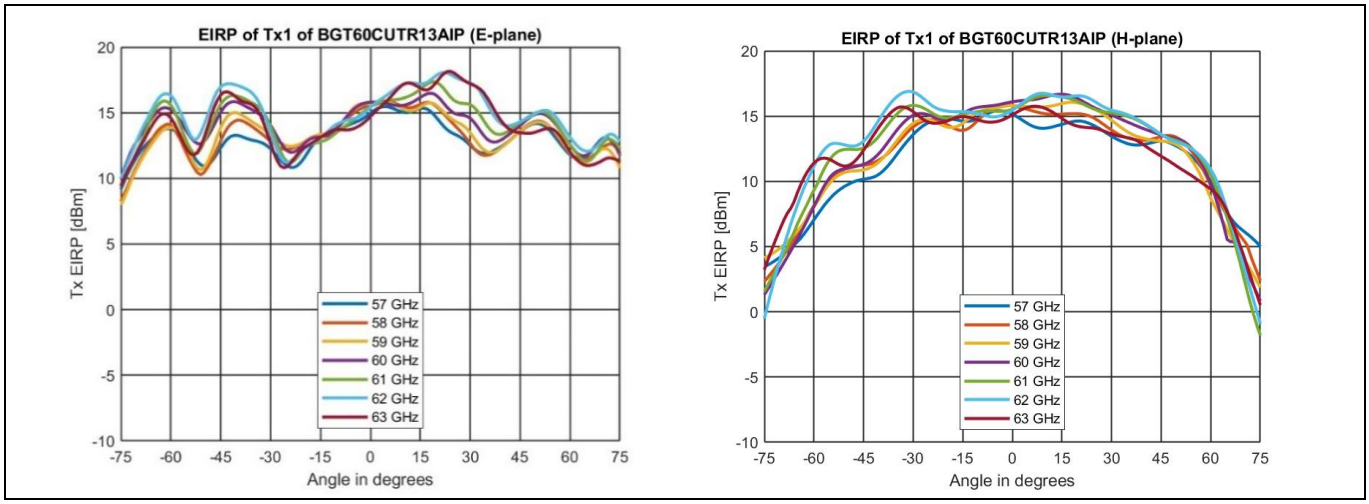


Figure 150 EIRP plot of Tx antenna of XENSIV™ BGT60CUTR13AIP V3 shield

10.7.4 Rx characterization

The Rx characterization measurement aims to obtain the gain level of the receive antennas of the radar system. The measurement setup consists of the following devices and measurement components, as shown in Figure 151:

- Device under test (DUT)
- Gimbal
- Horn antenna
- Antenna adapter
- Connecting cable
- Signal generator

The measurement process involves:

- **Providing a 0 dBm CW signal at the preferred center frequency using the signal generator:** A CW signal of 0 dBm is provided at the preferred center frequency using the signal generator to simulate the signal received by the Rx antennas. This is done to simulate the actual reception scenario, where the radar system receives a signal from a target

- **Transmitting the signal using the horn antenna setup:** The signal is transmitted using the horn antenna setup to simulate the signal transmission and reception process. This is done to ensure that the measurement is performed in a controlled environment

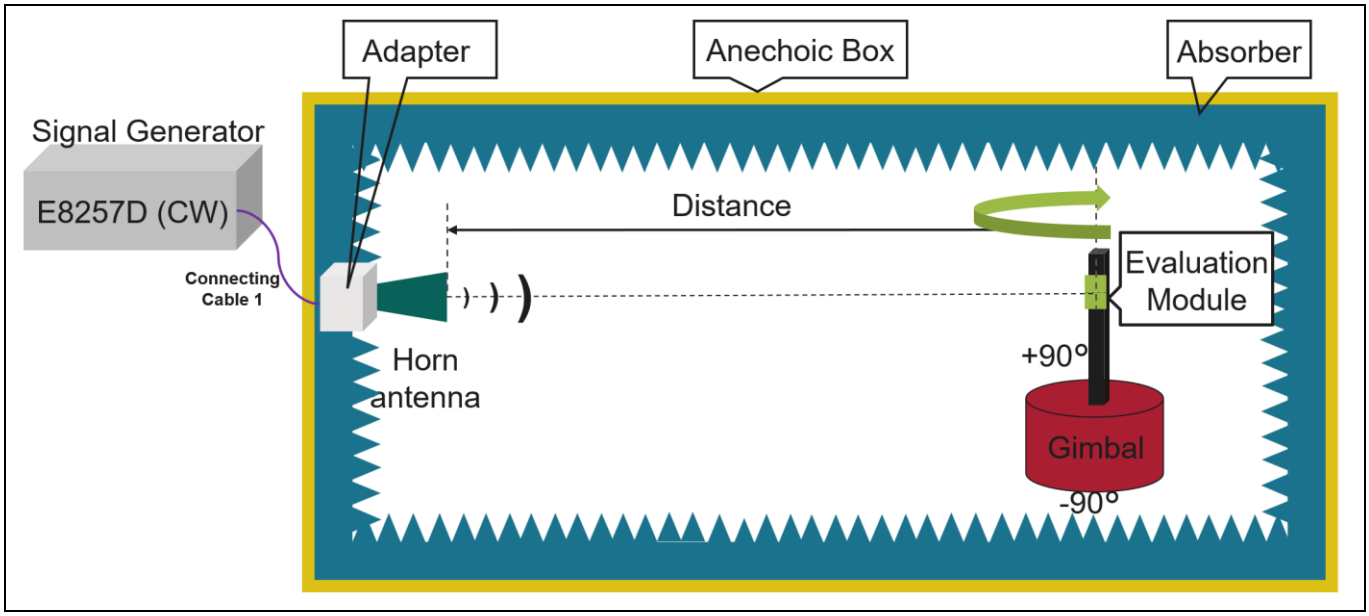


Figure 151 Measurement setup used to perform Rx characterization

- **Measuring the distance between the horn antenna and the DUT to account for free-space path loss:** The distance between the horn antenna and the DUT is measured to account for free-space path loss, which affects the signal strength and accuracy of the measurement
- **Enabling the Rx channels of the DUT to receive the signal and disabling the Tx channels to avoid crosstalk:** The Rx channels of the DUT are enabled to receive the signal, and the Tx channels are disabled to avoid crosstalk and ensure accurate measurement. This is done to ensure that the measurement is performed in a controlled environment
- **Compensating for cable losses, adapter losses, and free-space distance losses to obtain the gain value:** The losses incurred due to the connecting cable, antenna adapter, and free-space distance are compensated to obtain the accurate gain value. This is done to ensure that the measured gain value is accurate and representative of the radar system's performance

The Rx characterization measurement provides valuable information about the radar system's reception capabilities, including the gain level, which is critical for determining the system's sensitivity and accuracy.

10.7.4.1 Measurement results

Normalized RX gain plot in E- and H-plane from the Rx characterization measurement for BGT60CUTR13AIP V3 shield with respect to the maximum gain across all receivers and all CW frequencies from 57 GHz to 63 GHz in step of 1 GHz are shown in Figure 152 and Figure 153, respectively.

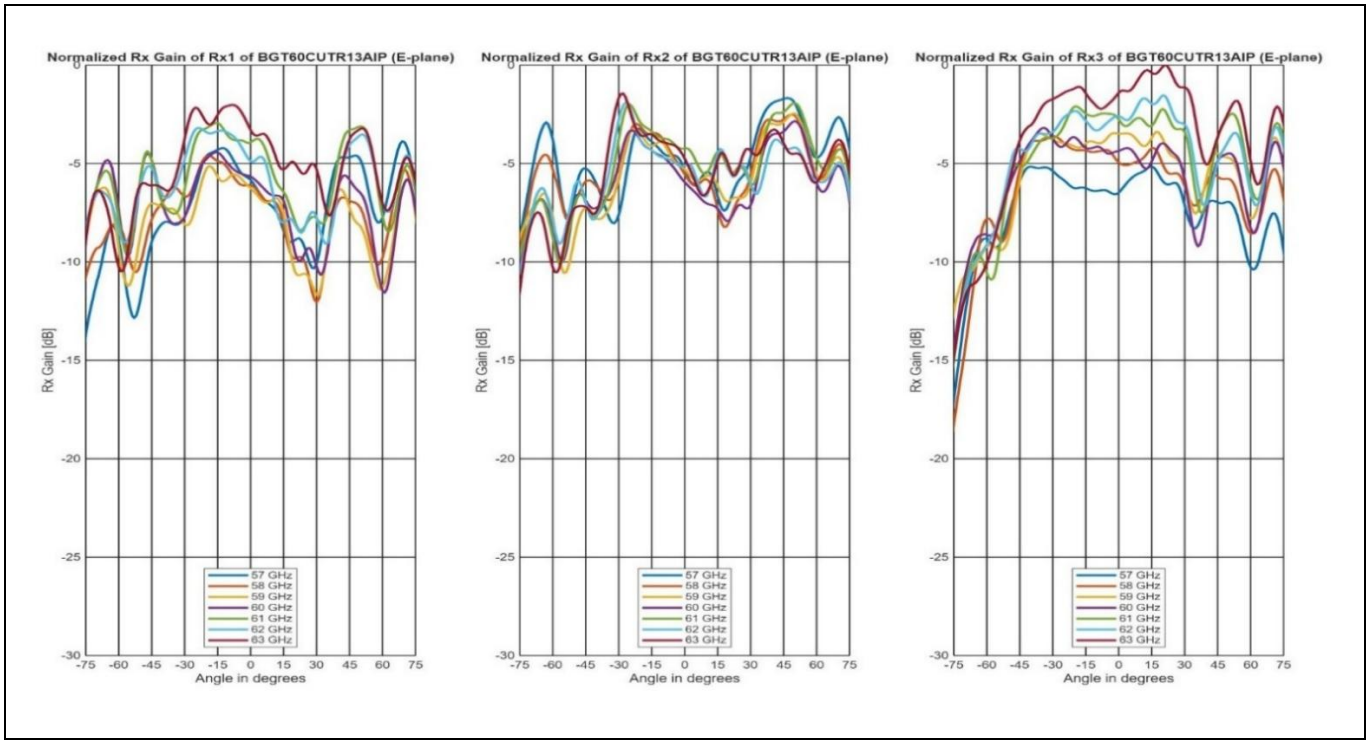


Figure 152 Normalized Rx Gain in E-plane with respect to maximum gain value across all receivers and all CW frequencies from 57 GHz to 63 GHz in step of 1 GHz

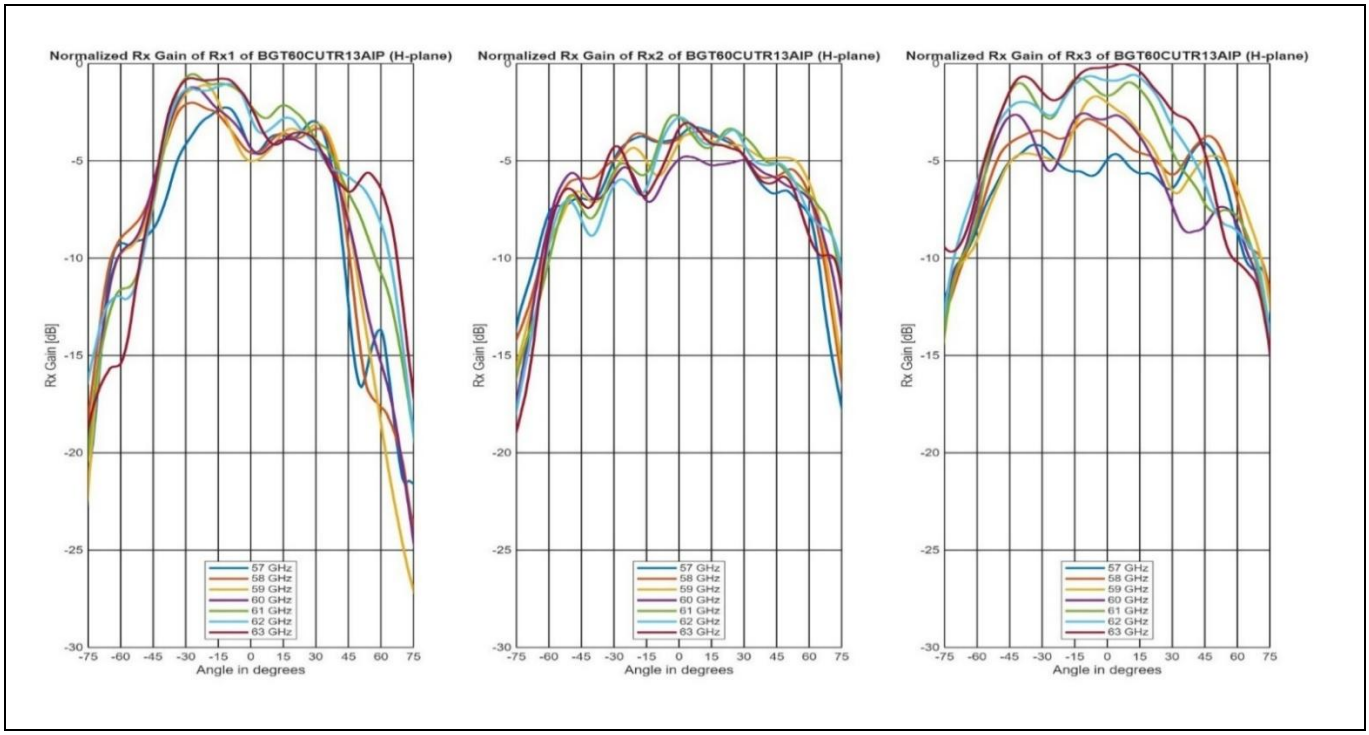


Figure 153 Normalized Rx Gain in H-plane with respect to maximum gain value across all receivers and all CW frequencies from 57 GHz to 63 GHz in step of 1 GHz

10.7.5 Radar mode measurement setup

The radar mode measurement aims to test the capability of the radar system to detect a target in FMCW mode. The measurement setup consists of the following devices and measurement components, as shown in Figure 154:

- Device under test (DUT)
- Gimbal
- Corner reflector

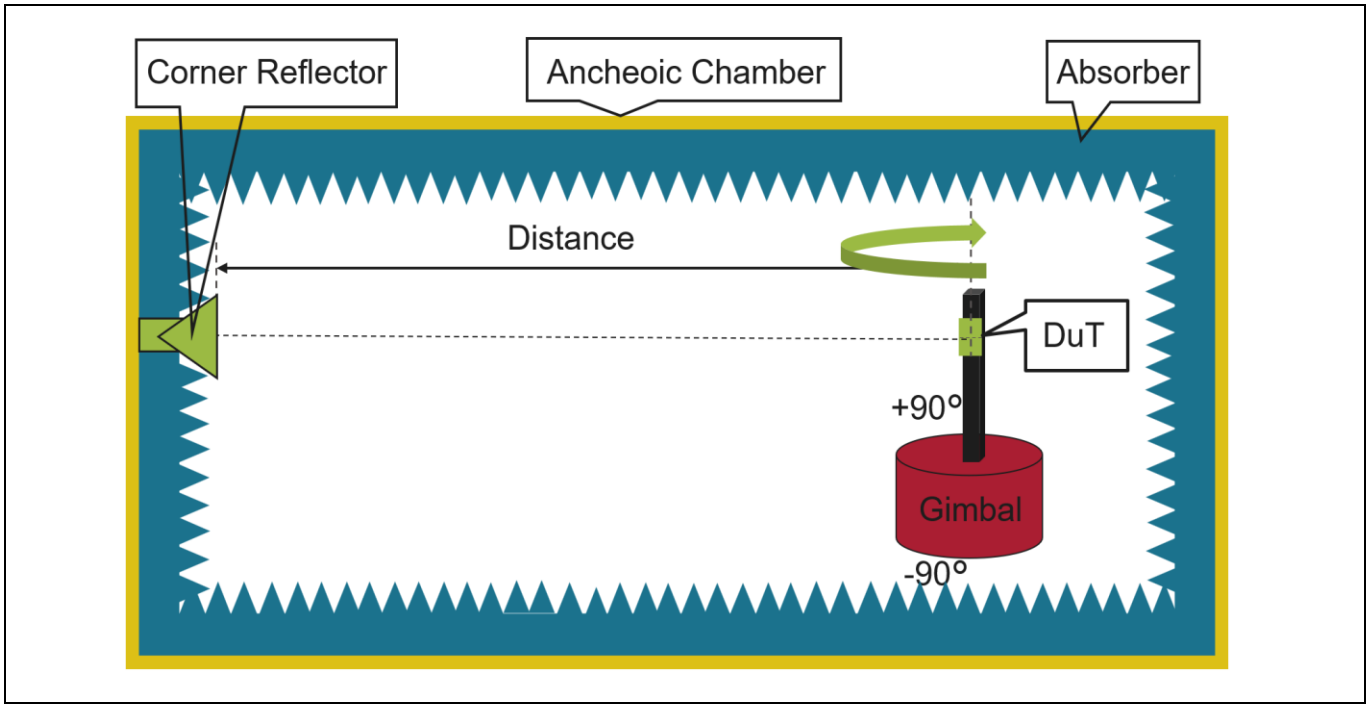


Figure 154 Anechoic chamber construction showing absorbers, corner reflector on left side and the DUT placed on the gimbal which is on right side

The measurement process involves:

- **Transmitting a chirp signal with a specific bandwidth using the DUT:** A chirp signal with a specific bandwidth is transmitted using the DUT to simulate the radar signal transmission. This is done to simulate the actual transmission scenario, where the radar system transmits a chirp signal to detect targets
- **Placing a corner reflector at a specific distance from the DUT:** A corner reflector is placed at a specific distance from the DUT to simulate the target reflection. This is done to ensure that the measurement is performed in a controlled environment
- **Receiving the reflected signal using the Rx antennas of the DUT:** The reflected signal is received using the Rx antennas of the DUT to simulate the signal reception process
- **Saving the raw ADC data for post-processing:** The raw ADC data is saved for post-processing to obtain the range-Doppler map, as shown in the Figure 155
- **Performing fast Fourier transform (FFT) on each chirp data to obtain the range-FFT visualization:** FFT is performed on each chirp data to obtain the range-FFT visualization, which provides information about the target's range
- **Performing FFT across the chirps to obtain the Doppler information:** FFT is performed across the chirps to obtain the Doppler information, which provides information about the target's velocity

- **Combining the range-FFT and Doppler-FFT processed data to obtain the range-Doppler (RD) map:** The range-FFT and Doppler-FFT processed data are combined to obtain the RD map, which provides information about the target's range and velocity

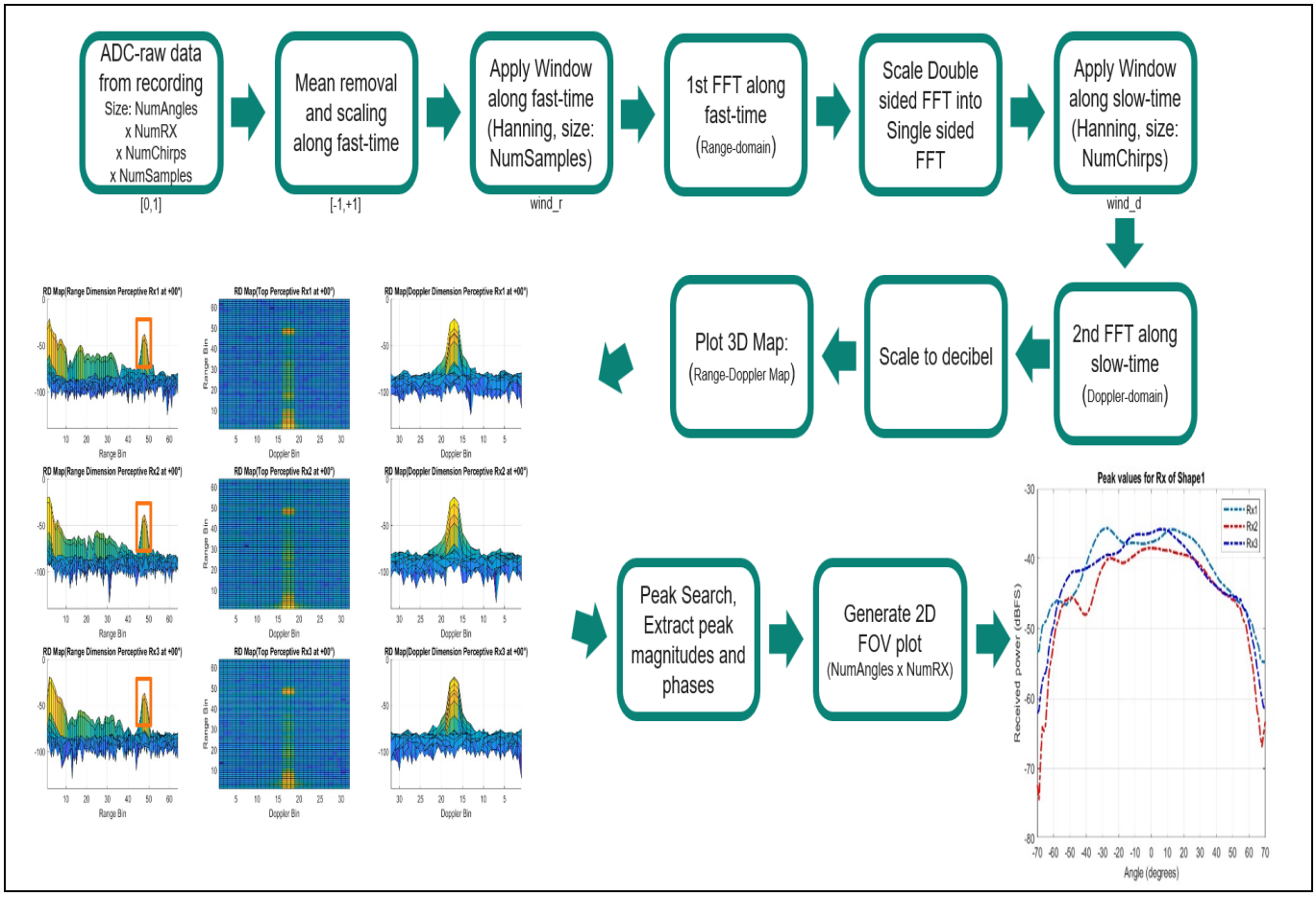


Figure 155 Post-processing of raw ADC data

- **Plotting the peak target power level across angles observed in the RD map to obtain the outcome of the radar mode measurement:** The peak target power level is plotted across angles observed in the RD map to obtain the outcome of the radar mode measurement, which provides information about the radar system's ability to detect targets at different angles. Like power level measured at every angle, noise floor level can also be measured. The measured noise floor level at every angle can be subtracted from power level at every angle to obtain the SNR level across angles

The Radar mode measurement provides valuable information about the radar system's ability to detect targets in FMCW mode, including the range-Doppler map and the peak target power level. This information is critical for determining the system's performance and accuracy in real-world scenarios.

10.7.5.1 Measurement results

Radar mode measurement result for XENSIV™ BGT60CUTR13AIP V3 shield performed for the bandwidths 61 GHz to 62 GHz, 60 GHz to 62 GHz, and 58 GHz to 63 GHz are shown in Figure 156, Figure 157, and Figure 158, respectively.

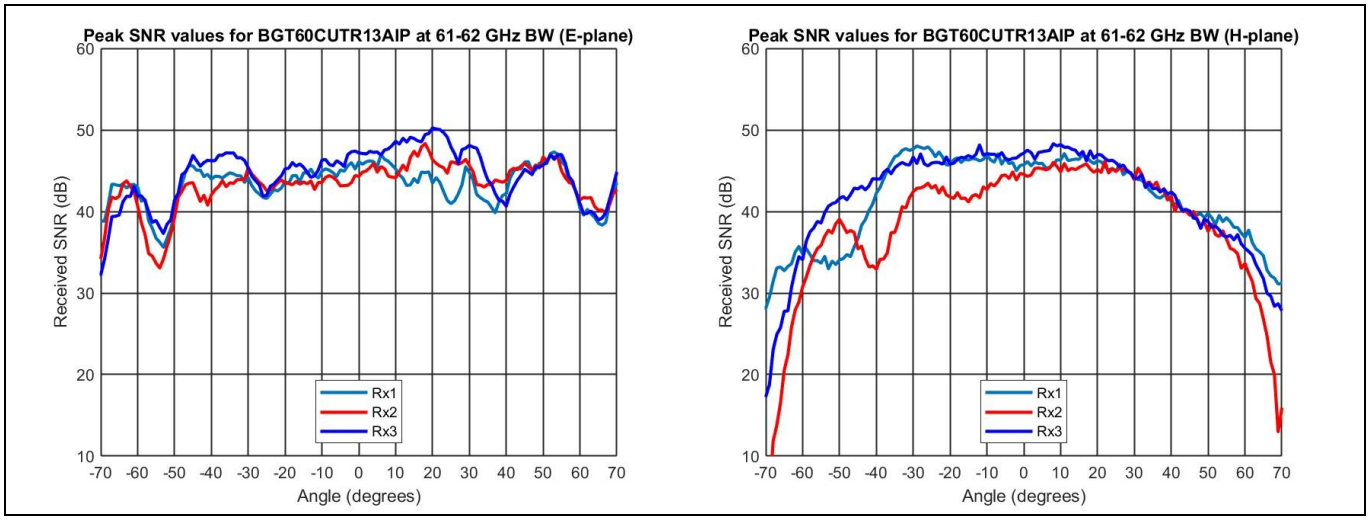


Figure 156 Radar mode measurement result for 61 GHz to 62 GHz bandwidth

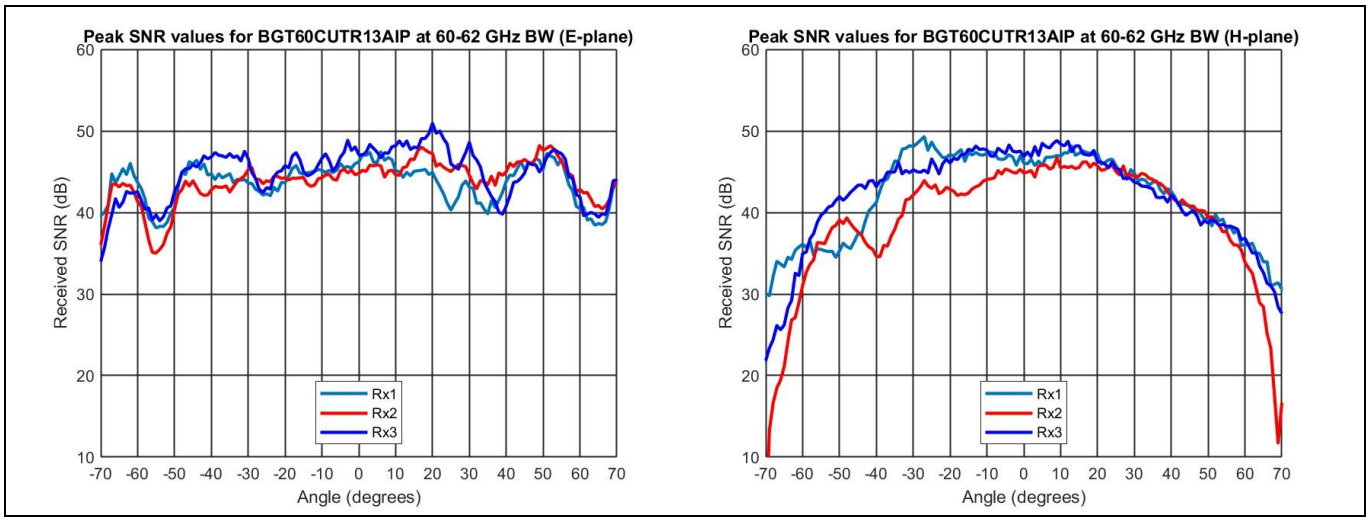


Figure 157 Radar mode measurement result for 60 GHz to 62 GHz bandwidth

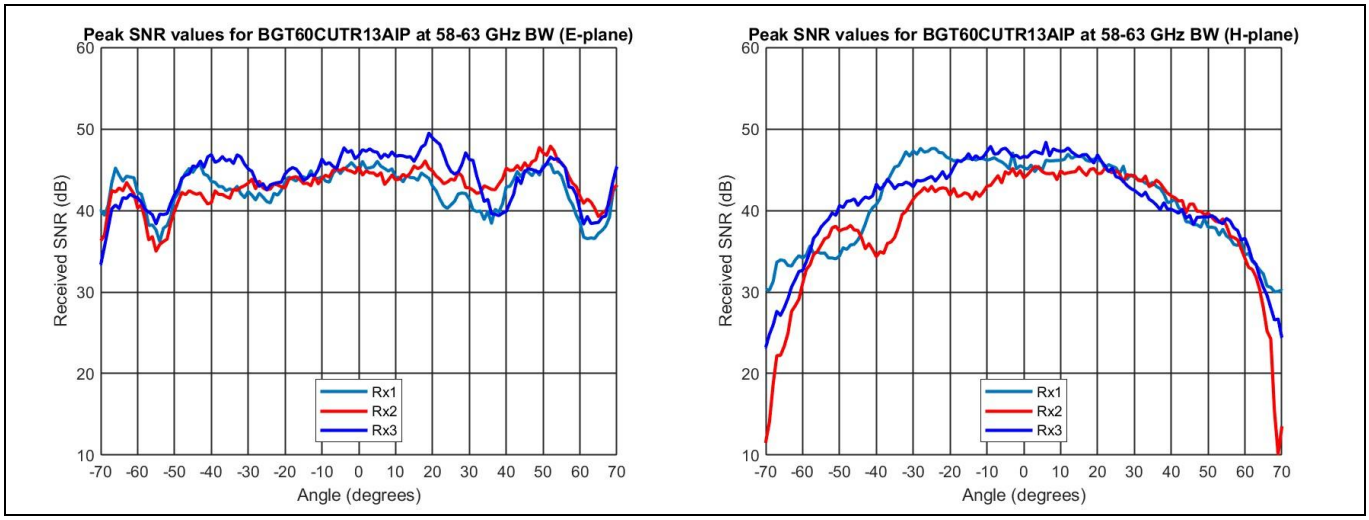


Figure 158 Radar mode measurement result for 58 GHz to 63 GHz bandwidth

11 Related resources

- Webpage: XENSIV™ 60 GHz CMOS radar sensors for IoT
- Infineon Developer Community: XENSIV™ radar sensors forum

References

[1] Infineon Technologies AG: *BGT60CUTR13AIP MMIC datasheet*; Available online

Glossary

AAF
anti-aliasing filter
A filter used before a signal sampler to restrict the bandwidth of a signal to satisfy the Nyquist–Shannon sampling theorem over the band of interest.

ABB
analog base band
A signal with a near-zero frequency range, this is, a spectral magnitude that is nonzero only for frequencies in the vicinity of the origin and negligible elsewhere.

AC
alternating current
A form of power supply in which the flow of electric charge periodically reverses direction.

ADC
analog-to-digital converter
A device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude.

AoA
angle of arrival

AP
application processor
A chip that is used to run primary user applications.

BF
register bitfield

CDM
charge device model
A model for characterizing the susceptibility of an electronic device to damage from electrostatic discharge.

CORDIC
coordinate rotation digital computer

CMOS
complementary metal-oxide semiconductor
A technology for constructing integrated circuits that uses a combination of p-channel and n-channel metal–oxide– semiconductor field-effect transistors (MOSFETs) to implement logic gates.

CUT
cell under test

CW
continuous wave
The emission of a sinusoidal radio wave, which may be modulated by signals.

DC
direct current
A form of power supply in which the flow of electric charge is only in one direction.

FIFO
first in first out
A method for organizing the manipulation of a data structure where the first entry is processed first.

FM
frequency-modulated
An angle modulation in which the instantaneous frequency deviation varies in accordance with a given function, generally linear, of the instantaneous value of the modulating signal.

FMCW
frequency-modulated continuous wave
A type of radar system where a known frequency of continuous wave radio energy is transmitted and then received from any reflecting object.

FoV
field of view

FSM
finite state machine
An abstract machine that can be in exactly one of a finite number of states at any given time.

HPF
high-pass filter
An electronic filter that passes signals with a frequency higher than a certain cutoff frequency and attenuates signals with frequencies lower than the cutoff frequency.

HW
hardware
The collection of physical components that comprise a computer or any other electronic system.

IF
intermediate frequency
The frequency corresponding to the carrier frequency or another characteristic frequency of an input radio-frequency signal in a signal resulting from each frequency translation.

LDO
low dropout voltage regulator
A direct current linear voltage regulator that can regulate the output voltage even if the supply voltage is very close to the output voltage.

LFSR
linear feedback shift register
A shift register whose input bit is a linear function of its previous state.

LO
local oscillator
An electronic oscillator used with a mixer to change the frequency of a signal.

LSB
least significant bit
The binary digit of the least significance within a code word.

MCU
microcontroller unit

A small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.

MISO
master in slave out
The input line of a Serial Peripheral Interface master.

MMIC
monolithic microwave integrated circuit

MOSI
master out slave in
The output line of a Serial Peripheral Interface master.

MSB
most significant bit
The binary digit of the most significance within a code word.

PA
power amplifier
An electronic circuitry that converts a low-power signal to a high-power signal.

PLL
phase-locked loop
A feedback circuit for synchronizing an oscillator with the phase of an input signal.

RF
radio frequency
A frequency of a periodic radio wave or of the corresponding electric oscillation.

RSVD
reserved
A placeholder for future functionalities or features.

RX
receiver
A device that accepts signals from remote transmitters.

SAR
successive approximation register
A type of analog-to-digital converter that converts a continuous analog waveform into a discrete digital representation using a binary search through all possible quantization levels.

SPI
Serial Peripheral Interface
A synchronous serial communication interface specification used for inter-chip communication, primarily in embedded systems.

SW
software
The collection of non-physical components that comprise a set of instructions executed by a processor.

TX
transmitter
A device that sends out signals.

VGA
variable gain amplifier
An electronic amplifier that varies its gain depending on a programmable control voltage.

Revision history

Document revision	Date	Description of changes
0.01	2024-06-03	• Initial version • Chapter Pin configuration added • Chapter Interfaces added
0.02	2024-06-11	• Chapter Device initialization added
0.03	2024-07-05	• Chapter Functional block description added
0.04	2024-07-11	• Chapter Radar frame definition added • Chapter Data flow added • Chapter Presence detection (started)
0.05	2024-07-18	• Chapter Presence detection added
0.10	2024-07-22	• Version with all major chapters included
0.11	2024-08-06	• Minor corrections • Chapter Angle estimation updated • Chapter Phase shifter added • Chapter Extended bandwidth (interlaced) chirps added • Chapter Infinite impulse response mode removed
0.12	2024-10-18	• Fixed typo in SPI chapter (LSB first to MSB first) • Updated PLL ramp figure and ramp formula • Replaced TX output power figure of TR23 by TR13 • Added clock configuration chapter
0.13	2024-11-12	• Angle estimation and calibration matrix sections are updated • Updated SPI timing diagram (High-Z state)
0.14	2025-01-30	• Chapter RF shield design, simulation and measurement added
0.20	2025-05-15	• Corrected typos and grammar • Rework of major chapters • Updated IO mapping descriptions for DMUX (MD sideband signals) • Added CFAR configuration example figure to CFAR chapter • Moved Presence Detection to separate chapter • Updated Boot Loader section • Updated Clock Configuration • Added Quick Start chapter • Removed MTI IIR mode as only MTI delta mode is used for MD/PD • Chapter on radome design guidelines with Simulations results • Updated incorrect SPI dual and Quad SPI figures
0.21	2025-07-07	• Corrected range zones from 9 to 16 in range angle zone masking • Corrected numbering in the range angle zone • MD_TARGET#_MAG scaling factor information is added

Revision history

Document revision	Date	Description of changes
		• Corrected T_FSTDN register name
0.30	2026-05-25	• ROM configuration chapter is added • Presence detection chapter is updated • TX, RX, and radar mode measurement plots are updated based on B-step silicon
0.40	2026-07-28	• Presence detection chapter is updated • RF shield design, simulation, and measurement chapter is updated
0.50	2026-11-19	• RF shield design, simulation, and measurement chapter is updated

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

Edition: 2026-11-19

Published by
Infineon Technologies AG
Am Campeon 1-15
85579 Neubiberg, Germany

© 2026 Infineon Technologies AG.
All rights reserved

Do you have a question about this document?
Email: erratum@infineon.com

Document reference:
UG143920

Important Notice

This evaluation board, demonstration board, reference board or kit and any related documentation (“Evaluation Board”) is sold or provided and delivered by Infineon Technologies AG and its affiliates (“Infineon”). Evaluation Boards are part of Infineon’s non-serial products.

This Evaluation Board is delivered to customer "AS IS", without any warranty (whether express or implied) or liability of Infineon of any kind, including but without limitation any warranty of merchantability, fitness for a particular purpose or non-infringement. These limitations and exclusions shall not apply in case of Infineon's willful misconduct or in any other case where liability is mandatory at law.

This Evaluation Board shall only be used by customer for test purposes and not for series production. However, the Evaluation Board shall not to be used for reliability testing. The Evaluation Board is exclusively intended for technically qualified and skilled customer representatives. The Evaluation Board shall not be used in any way where a failure of the Evaluation Board, or any consequences of the use thereof, can reasonably be expected to result in personal injury.

Evaluation Boards may not comply with CE or similar standards (including without limitation the EMC Directive 2004/EC/108 and the EMC Act) and may not fulfill other requirements of the country in which they are operated by the customer. The customer shall ensure that each Evaluation Board will be handled in a way which is compliant with all relevant requirements and standards in the country in which it is operated.

Software and associated documentation and materials included or referenced in the Evaluation Board (“Software”) is owned by Infineon and is protected by and subject to worldwide patent protection, worldwide copyright laws, and international treaty provisions. Therefore, you may use this Software only as provided in the license agreement accompanying the software package from which you obtained this Software.

If no license agreement applies, then any use, reproduction, modification, translation, or compilation of this Software is prohibited without the express written permission of Infineon. Unless otherwise expressly agreed with Infineon, this Software is provided to customer "AS IS", without any warranty (whether express or implied) or liability of Infineon of any kind, including but without limitation any warranty of merchantability, fitness for a particular purpose or non-infringement.

These limitations and exclusions shall not apply in case of Infineon's willful misconduct or in any other case where liability is mandatory at law. Infineon reserves the right to make changes to the Software without notice. You are responsible for properly designing, programming, and testing the functionality and safety of your intended application of the Software, as well as complying with any legal requirements related to its use. Infineon does not guarantee that the Software will be free from intrusion, data theft or loss, or other breaches (“Security Breaches”), and Infineon shall have no liability arising out of any Security Breaches. The Software shall not be used in any way where a failure of the Evaluation Board or any consequences of the use thereof, can reasonably be expected to result in personal injury.

Safety & Operating Instructions:

Customer shall check the Evaluation Board for any physical damage which may have occurred during transport. If customer detects any damages or defects in the Evaluation Board, customer shall not connect the Evaluation Board to a power source. Customer shall contact Infineon for further support. If customer observes unusual operating behavior during the evaluation process, customer shall immediately shut off the power supply to the Evaluation Board and consult Infineon for support.

Customer shall not touch the Evaluation Board during operation and keep a safe distance.

Customer shall not touch the Evaluation Board after disconnecting the power supply, several components may still store electrical voltage and can discharge through physical contact. Several parts, like heat sinks and transformers, may still be very hot. Allow the components to cool before touching or servicing.

The electrical installation must be completed in accordance with the appropriate safety requirements.