Lime Microsystems

Surrey Tech Centre
Occam Road
The Surrey Research Park
Guildford, Surrey GU2 7YG
United Kingdom

Tel: +44 (0) 1483 685 063
Fax: +44 (0) 1428 656 662
e-mail: enquiries@limemicro.com

LMS7002Mr2 – MULTI-BAND, MULTI-STANDARD MIMO RF TRANSCEIVER IC

- *SPI Register Map* -
(Version 01, Revision 18)

Chip version: LMS7002Mr2
Chip revision: 01
Document version: 01
Document revision: 18
Last modified: 26/09/2014 07:39:00

Contents

Revision History

Version 01r00
Started: 06 Dec, 2013
Finished: 06 Dec, 2013
Initial version.
Constructed from LMS7002M-SPIRegs-2.00r17.doc and
LMS7002M-SPIRegs-2.00r14_kk.doc documents.
Default values changed.

Version 01r01
Started: 10 Dec, 2013
Finished: 10 Dec, 2013
RxTSP and TxTSP description updated.

Version 01r02
Started: 11 Dec, 2013
Finished: 11 Dec, 2013
Document updated.
SXR, SXT Configuration Memory section changed.

Version 01r03
Started: 12 Dec, 2013
Finished: 12 Dec, 2013
Widths of following registers updated: C_CTL_PGA_RBB_(1, 2)[7:0], RESRV_TRF_(1, 2)[4:0], RESRV_TBB_(1, 2)[7:0], RESRV_CGN[3:1].

Version 01r04
Started: 12 Dec, 2013
Finished: 20 Dec, 2013
New controls (EN_DIR_) in 0x081 and 0x0124 registers.

EN_SDM_TSTO_CGEN moved from address 0x0086[7] to address 0x00A8[4].
EN_SDM_TSTO_(SXR, SXT) moved from address 0x011C[7] to address 0x00A8[5,6].
Added PD_FBDIV_(SXR, SXT) to address 0x011C[7].
Changed PD_FDIV_(SXR, SXT) description in address 0x011C.

Added CDC_I_TRF_(1,2)[3:0] and CDC_Q_TRF_(1,2)[3:0] to address 0x0104.
Address 0x0104 default changed.

Added CDC_I_RFE_(1,2)[3:0] and CDC_Q_RFE_(1,2)[3:0] to address 0x010C.
Address 0x010C default changed.

Version 01r05
Started: 30 Dec, 2013
Finished: 30 Dec, 2013

(*RESRV outputs use updated in next document revision*)
Removed RESRV_RFE_(1, 2) from address 0x0114.

Outputs used for :
CDC_I_RFE_(1, 2)[3:0], CDC_Q_RFE_(1, 2)[3:1] (controlling from address 0x010C)

Removed RESRV_TRF_(1, 2)[4:0] from address 0x0104
Outputs used for: CDC_I_TRF_2[3:0], CDC_Q_TRF_2[3]

RESRV_(SXR, SXT)[7:0] output width narrowed in address 0x0124.
SXT Outputs used for:
CDC_Q_TRF_1[2:0], CDC_Q_TRF_2[2:0], PD_FBDIV_SXT
SXR for:
CDC_Q_RFE_1[0], CDC_Q_RFE_2[0], PD_FBDIV_SXR

Version 01r06
Started: 06 Jan, 2014
Finished: 06 Jan, 2014

Removed RESRV_RFE_(1, 2) from address 0x0114.
Outputs used for:
CDC_I_RFE_(1, 2)[3:0], CDC_Q_RFE_(1, 2)[3:1] (controlling from address 0x010C)

Removed RESRV_TRF_(1, 2)[4:0] from address 0x0104.
RESRV_TRF_1 outputs used for: CDC_I_TRF_1[3:2], CDC_I_TRF_2[3:1].
RESRV_TRF_2 outputs used for: CDC_I_TRF_2[0], CDC_Q_TRF_2[3:0].

RESRV_(SXR, SXT)[7:0] output width narrowed in address 0x0124.
RESRV_SXT outputs used for: CDC_I_TRF_1[1:0], CDC_Q_TRF_1[3:0], PD_FBDIV_SXT.
RESRV_SXR outputs used for: CDC_Q_RFE_2[0], CDC_Q_RFE_1[0], PD_FBDIV_SXR.

Version 01r07
Started: 06 Jan, 2014
Finished: 06 Jan, 2014

LML_TXNRXIQ1 default changed (address 0x0023[1]).
New registers VER[4:0], REV[4:0], MASK[5:0] added, address 0x002F.
EN_DIR_BIAS register removed (0x0081).
EN_BIAS_MASTER register name changed to PD_BIAS_MASTER (0x0084).
PD_* registers moved one bit to the left (0x0084).
EN_G_BIAS register removed (0x0084).

Version 01r08
Started: 08 Jan, 2014
Finished: 08 Jan, 2014

Default values in registers at 0x0104 and 0x010C changed.

Version 01r09
Started: 10 Jan, 2014
Finished: 10 Jan, 2014

Default value of register at 0x002A changed.
LimeLight control structure diagram updated.
New register RXP_TST added to 0x002A[8].

Version 01r10
Started: 20 Jan, 2014
Finished: 20 Jan, 2014

Default values of registers at 0x0108, 0x0109, 0x010A, 0x0082, 0x0120, 0x0121, 0x009E addresses were changed.
Shadow registers added to CGEN module for registers:
　　　DIV_OUTCH_CGEN[7:0] (0x0089[10:3]);
　　　CSW_VCO_CGEN[7:0] (0x008B[8:1]).
Shadow registers added to SXT/SXR modules for registers:
　　　EN_DIV2_DIVPROG_(SXR, SXT) (0x011C[10]);
　　　DIV_LOCH_(SXR, SXT) [2:0] (0x011F[8:6]);
　　　SEL_VCO_(SXR, SXT)[1:0] (0x0121[2:1]);
　　　CSW_VCO_(SXR, SXT)[7:0] (0x0121[10:3]).
Shadow registers added to NCO control modules for registers:
　　　SEL[3:0] (0x0240[4:1] / 0x0440[4:1]);
　　　MODE (0x0240[0] / 0x0440[0]).

Version 01r11
Started: 21 Jan, 2014
Finished: 27 Jan, 2014

New registers added:
　　　NCO_BYP (0x0208[9]);
　　　DC_REG[15:0] (0x020C[15:0] / 0x040B);
　　　INSEL (0x0200[2] / 0x0400[2]);
　　　TSGMODE (0x0200[3] / 0x0400[3]);
　　　TSGSWAPIQ (0x0200[4] / 0x0400[4]);
　　　TSGDCLDI (0x0200[5] / 0x0400[5]);
　　　TSGDCLDQ (0x0200[6] / 0x0400[6]);
　　　TSGFCW (0x0200[8:7] / 0x0400[8:7]);
　　　CAPSEL[1:0] (0x0400[3:2]);
　　　CAPTURE (0x0400[4]);
　　　CAPD[31:0].
Registers removed:
　　　RSSI[15:0] (0x040B[15:0]);
　　　BSIGI[22:0], BSIGQ[22:0] and BSTATE (0x040D[15:0], 0x040E[15:0], 0x040F[15:0]).

Version 01r12
Started: 11 Feb, 2014
Finished: 18 Feb, 2014

New registers added:
        TSGFC (0x0200[9] / 0x0400[9]).
Register CAPTURE (0x0400[11]) mooved to 0x0400[15].
Register CAPSEL[1:0] (0x0400[10:9]) mooved to 0x0400[14:13].
Default value of 0x0400 updated.
Register MCLK1_DIV[7:0] renamed to RXTSPCLKA_DIV; register MCLK2_DIV[7:0] renamed to TXTSPCLKA_DIV (address 0x002C).
Default value of RDIV_VCOGN[7:0] (0x009A[7:0]) changed.

Version 01r13
Started: 17 Mar, 2014
Finished: 17 Mar, 2014

Description of register CAPD[31:0] (addresses 0x040E and 0x040F) updated.

Version 01r14
Started: 02 Apr, 2014
Finished: 14 Apr, 2014

Register CMIX_P6DB (addresses 0x0208 and 0x040C) is 2 bit instead of 1 and renamed to CMIX_GAIN.
Register CMIX_SC (0x0208[14] and 0x040C[14]) mooved to 0x0208[13] and 0x040C[13].
Register NCO_BYP removed (0x0208[9]).
NCO dithering control implemented (new register added to 0x0240/0x0440 addresses).
EN_DACCLKH_CLKGN renamed to EN_ADCCLKH_CLKGN (address 0x0086).
RESRV_TBB_(1, 2) resized from 8 bits to 6 (address 0x010B).
Default value updated in C_CTL_LPFH_RBB_(1, 2)[7:0] register description (0x0116[7:0]).
Default value updated in RDIV_VCOGN [7:0] register description (0x009A[7:0]).
Default value updated for CCOMP_TIA_RFE_(1, 2)[3:0] register (0x0112[15:12]).
Default value updated for RCOMP_TIA_RFE_(1, 2)[3:0] register (0x0114[8:5]).
Register RX_MUX[1:0] added instead of DLB_MUX (address 0x002A[7:8]).

Version 01r15
Started: 17 Apr, 2014
Finished: 19 Apr, 2014

Register RXRDCLK_MUX increased to 2 bits: RXRDCLK_MUX[1:0] (0x002A[3:2]).
Register TXWRCLK_MUX moved from 0x002A[3] to 0x002A[4].
Register TXRDCLK_MUX[1:0] moved from 0x002A[5:4] to 0x002A[6:5].
Register TX_MUX moved from 0x002A[6] to 0x002A[7].
Register RX_MUX[1:0] moved from 0x002A[8:7] to 0x002A[9:8].
Default of 0x002A changed.
New registers FCLK2_INV (0x002B[15]) and FCLK1_INV (0x002B[14]) added.
New registers MCLK2DLY[1:0] (0x002B[13:12]) and MCLK1DLY[1:0] (0x002B[11:10]) added.

Version 01r16
Started: 20 Apr, 2014
Finished: 20 Apr, 2014

New registers DIQDIRCTR2 (0x0023[15]), DIQDIR2 (0x0023[14]) DIQDIRCTR1 (0x0023[13]), DIQDIR1 (0x0023[12]) added.
New registers ENABLEDIRCTR2 (0x0023[11]), ENABLEDIR2 (0x0023[10]) ENABLEDIRCTR1 (0x0023[9]), ENABLEDIR1 (0x0023[8]) added.
TXWRCLK_MUX and TX_MUX registers resized to 2 bits (address 0x002a), default of 0x002A changed, registers in this address rearanged.

Version 01r17
Started: 31 Jul, 2014
Finished: 31 Jul, 2014

Description of register TST_SX_(SXR, SXT)[2:0] (address 0x011F) updated.
Description of register TST_CGEN[2:0] (address 0x0089) updated.
Register EN_INSHSW_H_RFE_(1, 2) removed (address 0x010D[5]).
All the control diagrams of analog blocks updated.
LML block diagram updated.
TxTSP block diagram updated.
RxTSP block diagram updated.

Version 01r18
Started: 07 Sep, 2014
Finished: 07 Sep, 2014

Description of register FRAC_SDM_(SXR, SXT) (address 0x011D) updated.

Version 2 2r0
Started: 25 Sep, 2014
Finished: 25 Sep, 2014

Final proofing, minor corrections and formatting

1

Serial Port Interface

1.1 Description

The functionality of the LMS7002Mr2 transceiver is fully controlled by a set of internal registers which can be accessed through a serial SPI port interface. Both write and read operations are supported. The serial SPI port can be configured to run in 3 or 4 wire mode with the following pins used:

- SEN   SPI serial port enable, active low, output from master;
- SCLK   SPI serial clock, output from master;
- SDIO   SPI serial data in/out (Master Output Slave Input (MOSI) / Master Input Slave Output (MISO)) in 3 wire mode, serial data input (MOSI) in 4 wire mode;
- SDO   SPI serial data out (MISO) in 4 wire mode, don't care in 3 wire mode.

SPI serial port key features:

- Operating as slave;
- Operating in SPI Mode 0 (data is captured on the clock's rising edge, while data is shifted on the clock's falling edge);
- 32 serial clock cycles are required to complete write operation;
- 32 serial clock cycles are required to complete read operation;
- Multiple write/read operations are possible without toggling serial enable signal.

All configuration registers are 16-bit wide. A write/read sequence consists of a 16-bit instruction followed by a 16-bit data to write or read. The MSB of the instruction bit stream is used as SPI command where CMD = 1 for write and CMD = 0 for read. The next 4 bits are reserved (Reserved[3:0]) and must be zeroes. The next 5 bits represent block address (Maddress[4:0]) since the LMS7002Mr2 configuration registers are divided into logical blocks as shown in Table 1. The remaining 6 bits of the instruction are used to address particular registers (Reg[5:0]) within the block as described in Section 2. Maddress and Reg compiles a global 11-bit register address when concatenated ((Maddress << 6) | Reg). Use global address values for a particular register from the tables provided in Section 2.

The write/read cycle waveforms are shown in *Figure 1*, *Figure 2* and *Figure 3*. Note that the write operation is the same for both 3-wire and 4-wire modes. Although not shown in the figures, multiple byte write/read is possible by repeating instruction/data sequence while keeping SEN low.
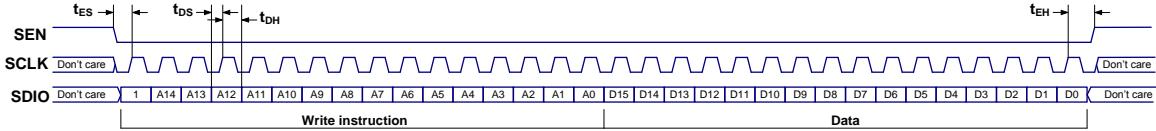
Figure 1 SPI write cycle, 3-wire and 4-wire modes
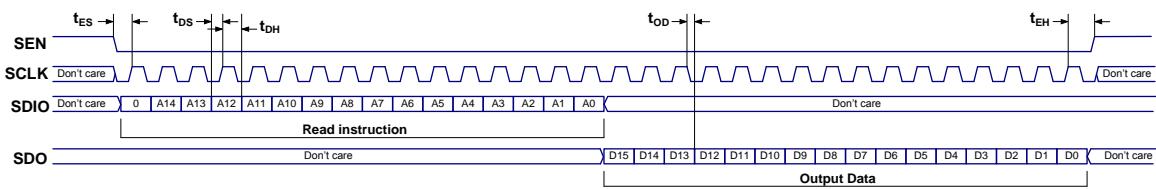
Figure 2 SPI read cycle, 4-wire mode (default)
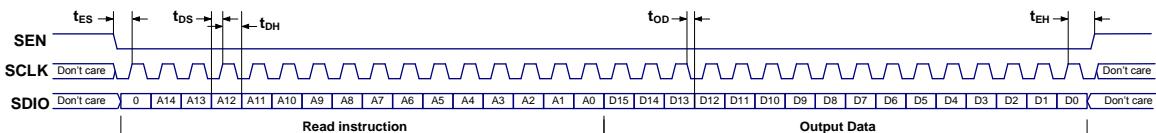
Figure 3 SPI read cycle, 3-wire mode

2

LMS7002Mr2 Memory Map Description

2.1 LMS7002Mr2 Memory Map

All the LMS7002Mr2 configuration space is accessible via the serial SPI interface. All the configuration space is divided to logical block types:

- Other
- Top
- TRX
- TX
- RX

The LMS7002Mr2 chip is MIMO, hence it has two channels called A and B. So, some analogue/digital modules appears in MIMO channel A as well as B (from TRX, TX and RX blocks). The rest of the modules (from Other and Top logical block types) are controlled only from one memory block. All the logical blocks are summarized in Table 1.

To save the addressing space and speed-up write operation the following technique can be used for the TRX, TX and RX logical block types. There is a register called MAC[1:0] (address of this register is 0x0020[1:0]) which selects MIMO channel A or/and B. MIMO channel select logic depends on MAC[1:0] register as described below (see *Figure 1* for reference):

- 11 – SPI write operation possible **only**. The same data patterns are written to the A and B MIMO channels at the same time. Note, that read operation will corrupt read data when MAC[1:0] is set to "11".
- 01 – SPI read/write operation possible. Data may be written to or read from the MIMO channel A only.
- 10 – SPI read/write operation possible. Data may be written to or read from the MIMO channel B only.

Using the MAC register simplifies programming for MIMO. As an example, the addresses of the registers controlling TBBA and TBBB are the same, but the individual A or B channels are identified using the MAC[1:0] register.

Let us consider the write operation to the G_TIA_RFE_A[1;0] register. This register controls the RFE module within MIMO channel A. To write to the G_TIA_RFE_A[1;0] register, we have to set MAC[1:0] to the "01". If we set MAC[1:0] to the "11" then the same value will be written to the registers G_TIA_RFE_A[1;0] and G_TIA_RFE_B[1;0] at the same time (i.e. only one write operation is required, hence time saved). Similarly, if we want to write to the G_TIA_RFE_B[1;0] register only, we have to set MAC[1:0] to "10".

The special case is frequency synthesizers SXR and SXT. Register addresses are the same for SXR and SXT. To control SXT we have to set MAC[1:0] to the "01" and MAC[1:0] to the "10" for SXR.

Modules from the Top and Other logical blocks (see Table 1) are not controlled by the MAC[1:0] register.
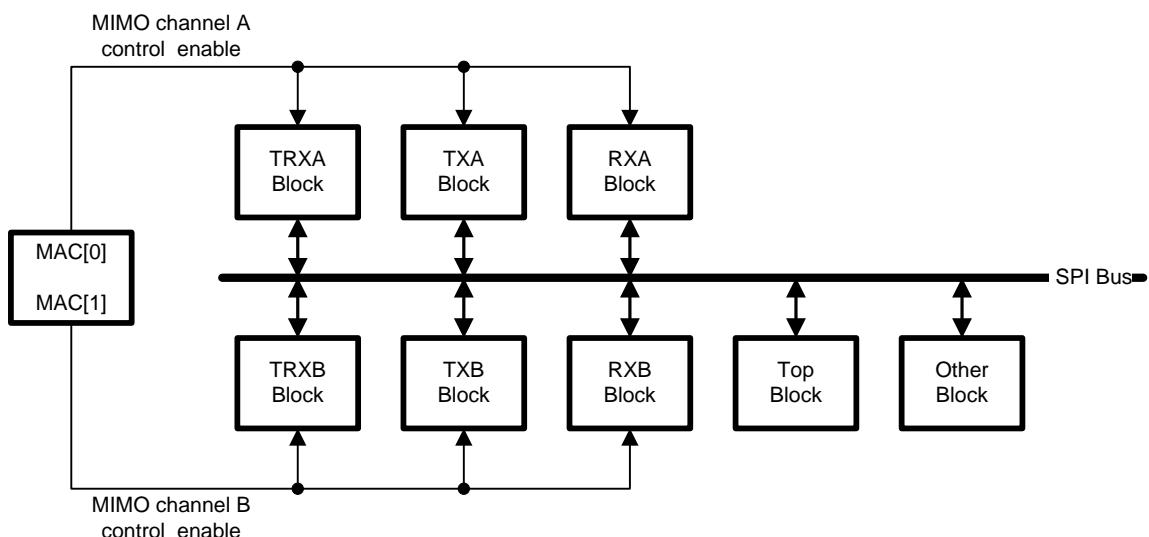
Figure 4 Access logic of configuration modules

The memory mapping is shown in Table 1. There are five basic logical blocks. These are:

a) Other, controlling the microcontroller and LimeLight^{TM} interface;
b) Top, controlling the top level bias, clock synthesizers, buffers, LDOs and BIST;
c) TRX, controlling the Transmit and Receive RF functions;
d) TX, controlling the transmit digital functions;
e) RX, controlling the receive digital functions.

Table 1: LMS7002Mr2 memory map

Logical Block Type	Logical Block Name	Size, regs	Cmd (R/W)	Address			Comments
				Resserved [3:0]	Maddress [4:0]	Reg [5:0]	
Other	uC	16	0/1	0000	00000	00xxxx	Address space starts at 0x0000. Addressing does not depend on MAC[1:0].
	Lime Light	32	0/1	0000	00000	1xxxxx	Address space starts at 0x0020. Addressing does not depend on MAC[1:0].
TOP	Top Control (AFE, BIAS, XBUF, CGEN, LDO, BIST)	128	0/1	0000	0001x	xxxxxx	Address space starts at 0x0080. Addressing does not depend on MAC[1:0].
TRX	TRX (TRF(A/B), TBB(A/B), RFE(A/B), RBB(A/B), SX(R/T)	128	0/1	0000	0010x	xxxxxx	Address space starts at 0x0100. Selected MIMO channel depends on MAC[1:0].
TX	TxTSP(A/B)	32	0/1	0000	01000	0xxxxx	Address space starts at 0x0200. Selected MIMO channel depends on MAC[1:0].
	TxNCO(A/B)	64	0/1	0000	01001	xxxxxx	Address space starts at 0x0240. Selected MIMO channel depends on MAC[1:0].
	TxGFIR1(A/B)	64	0/1	0000	01010	xxxxxx	Address space starts at 0x0280. Selected MIMO channel depends on MAC[1:0].
	TxGFIR2(A/B)	64	0/1	0000	01011	xxxxxx	Address space starts at 0x02C0. Selected MIMO channel depends on MAC[1:0].
	TxGFIR3a(A/B)	64	0/1	0000	01100	xxxxxx	Address space starts at 0x0300. Selected MIMO channel depends on MAC[1:0].
	TxGFIR3b(A/B)	64	0/1	0000	01101	xxxxxx	Address space starts at 0x0340. Selected MIMO channel depends on MAC[1:0].
	TxGFIR3c(A/B)	64	0/1	0000	01110	xxxxxx	Address space starts at 0x0380. Selected MIMO channel depends on MAC[1:0].
RX	RxTSP(A/B)	32	0/1	0000	10000	0xxxxx	Address space starts at 0x0400. Selected MIMO channel depends on MAC[1:0].
	RxNCO(A/B)	64	0/1	0000	10001	xxxxxx	Address space starts at 0x0440. Selected MIMO channel depends on MAC[1:0].
	RxGFIR1(A/B)	64	0/1	0000	10010	xxxxxx	Address space starts at 0x0480. Selected MIMO channel depends on MAC[1:0].
	RxGFIR2(A/B)	64	0/1	0000	10011	xxxxxx	Address space starts at 0x04C0. Selected MIMO channel depends on MAC[1:0].
	RxGFIR3a(A/B)	64	0/1	0000	10100	xxxxxx	Address space starts at 0x0500. Selected MIMO channel depends on MAC[1:0].
	RxGFIR3b(A/B)	64	0/1	0000	10101	xxxxxx	Address space starts at 0x0540. Selected MIMO channel depends on MAC[1:0].
	RxGFIR3c(A/B)	64	0/1	0000	10110	xxxxxx	Address space starts at 0x0580. Selected MIMO channel depends on MAC[1:0].

2.2 General Control, LimeLight^{TM} and IO Cell Configuration Memory

The block diagram of each IO cell is shown in *Figure 21*. It is possible to control the drive strength and pull-up resistor value of each IO cell.

The tables in this chapter describe the control registers of the IO cells and LimeLight^{TM} Ports 1 and 2. The control diagram of the LimeLight^{TM} ports is shown in Figure 27.

The general purpose control registers are also described in this chapter.

Table 2 LimeLight^{TM} and PAD configuration memory

Address (15 bits)	Bits	Description
0x0020	15	LRST_TX_B: Resets all the logic registers to the default state for Tx MIMO channel B. 0 – Reset active 1 – Reset inactive **(default)**
	14	MRST_TX_B: Resets all the configuration memory to the default state for Tx MIMO channel B. 0 – Reset active 1 – Reset inactive **(default)**
	13	LRST_TX_A: Resets all the logic registers to the default state for Tx MIMO channel A. 0 – Reset active 1 – Reset inactive **(default)**
	12	MRST_TX_A: Resets all the configuration memory to the default state for Tx MIMO channel A. 0 – Reset active 1 – Reset inactive **(default)**
	11	LRST_RX_B: Resets all the logic registers to the default state for Rx MIMO channel B. 0 – Reset active 1 – Reset inactive **(default)**
	10	MRST_RX_B: Resets all the configuration memory to the default state for Rx MIMO channel B. 0 – Reset active 1 – Reset inactive **(default)**
	9	LRST_RX_A: Resets all the logic registers to the default state for Rx MIMO channel A. 0 – Reset active 1 – Reset inactive **(default)**
	8	MRST_RX_A: Resets all the configuration memory to the default state for Rx MIMO channel A. 0 – Reset active 1 – Reset inactive **(default)**
	7	SRST_RXFIFO: RX FIFO soft reset (LimeLight^{TM} Interface). 0 – Reset active 1 – Reset inactive **(default)**
	6	SRST_TXFIFO: TX FIFO soft reset (LimeLight^{TM} Interface). 0 – Reset active 1 – Reset inactive **(default)**
	5	RXEN_B: Power control for Rx MIMO channel B. 0 – Rx MIMO channel B powered down 1 – Rx MIMO channel B enabled **(default)**
	4	RXEN_A: Power control for Rx MIMO channel A. 0 – Rx MIMO channel A powered down 1 – Rx MIMO channel A enabled **(default)**
	3	TXEN_B: Power control for Tx MIMO channel B. 0 – Tx MIMO channel B powered down 1 – Tx MIMO channel B enabled **(default)**
	2	TXEN_A: Power control for Tx MIMO channel A. 0 – Tx MIMO channel A powered down 1 – Tx MIMO channel A enabled **(default)**
	1 – 0	MAC[1:0]: MIMO access control. 11 – Channels A and B accessible. SPI write operation only (**default**) 01 – Channel A accessible only. Valid for SPI read/write 10 – Channel B accessible only. Valid for SPI read/write **Default**: 11111111 11111111

Address (15 bits)	Bits	Description
0x0021	15 – 12	**Reserved**
	11	TX_CLK_PE: Pull up control of TX_CLK pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	10	RX_CLK_PE: Pull up control of RX_CLK pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	9	SDA_PE: Pull up control of SDA pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	8	SDA_DS: Driver strength of SDA pad.
		0 – Driver strength is 4mA **(default)**
		1 – Driver strength is 8mA
	7	SCL_PE: Pull up control of SCL pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	6	SCL_DS: Driver strength of SCL pad.
		0 – Driver strength is 4mA **(default)**
		1 – Driver strength is 8mA
	5	SDIO_DS: Driver strength of SDIO pad.
		0 – Driver strength is 4mA **(default)**
		1 – Driver strength is 8mA
	4	SDIO_PE: Pull up control of SDIO pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	3	SDO_PE: Pull up control of SDO pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	2	SCLK_PE: Pull up control of SCLK pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	1	SEN_PE: Pull up control of SEN pad.
		0 – Pull up disengaged
		1 – Pull up engaged **(default)**
	0	SPIMODE: SPI communication mode.
		0 – 3 wire mode
		1 – 4 wire mode **(default)**
		Default: 00001110 10011111

Address (15 bits)	Bits	Description
0x0022	15 – 12	**Reserved**
	11	DIQ2_DS: Driver strength of DIQ2 pad. 0 – Driver strength is 4mA **(default)** 1 – Driver strength is 8mA
	10	DIQ2_PE: Pull up control of DIQ2 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	9	IQ_SEL_EN_2_PE: Pull up control of IQ_SEL_EN_2 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	8	TXNRX2_PE: Pull up control of TXNRX2 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	7	FCLK2_PE: Pull up control of FCLK2 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	6	MCLK2_PE: Pull up control of MCLK2 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	5	DIQ1_DS: Driver strength of DIQ1 pad. 0 – Driver strength is 4mA **(default)** 1 – Driver strength is 8mA
	4	DIQ1_PE: Pull up control of DIQ1 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	3	IQ_SEL_EN_1_PE: Pull up control of IQ_SEL_EN_1 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	2	TXNRX1_PE: Pull up control of TXNRX1 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	1	FCLK1_PE: Pull up control of FCLK1 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)**
	0	MCLK1_PE: Pull up control of MCLK1 pad. 0 – Pull up disengaged 1 – Pull up engaged **(default)** **Default**: 00000111 11011111

Address (15 bits)	Bits	Description
0x0023	15	DIQDIRCTR2: DIQ2 direction control mode. 0 – Automatic **(default)** 1 – Manual, controllable from DIQDIR2
	14	DIQDIR2: DIQ2 direction. 0 – Output 1 – Input **(default)**
	13	DIQDIRCTR1: DIQ1 direction control mode. 0 – Automatic **(default)** 1 – Manual, controllable from DIQDIR1
	12	DIQDIR1: DIQ1 direction. 0 – Output 1 – Input **(default)**
	11	ENABLEDIRCTR2: ENABLE2 direction control mode. 0 – Automatic **(default)** 1 – Manual, controllable from ENABLEDIR2
	10	ENABLEDIR2: ENABLE2 direction. 0 – Output 1 – Input **(default)**
	9	ENABLEDIRCTR1: ENABLE1 direction control mode. 0 – Automatic **(default)** 1 – Manual, controllable from ENABLEDIR1
	8	ENABLEDIR1: ENABLE1 direction. 0 – Output 1 – Input **(default)**
	7	**Reserved**
	6	MOD_EN: LimeLight™ interface enable. 0 – Interface disabled 1 – Interface enabled **(default)**
	5	LML_FIDM2: Frame start ID selection for Port 2, when LML_MODE2 = 0. 0 – Frame start, when 0 **(default)** 1 – Frame start, when 1
	4	LML_TXNRXIQ2: TXIQ/RXIQ mode selection for Port 2, when LML_MODE2 = 0. 0 – RXIQ mode 1 – TXIQ mode **(default)**
	3	LML_MODE2: Mode of LimeLight™ Port 2. 0 – TRXIQ mode 1 – JESD207 mode **(default)**
	2	LML_FIDM1: Frame start ID selection for Port 1, when LML_MODE1 = 0. 0 – Frame start, when 0 **(default)** 1 – Frame start, when 1
	1	LML_TXNRXIQ1: TXIQ/RXIQ mode selection for Port 1, when LML_MODE1 = 0. 0 – RXIQ mode **(default)** 1 – TXIQ mode
	0	LML_MODE1: Mode of LimeLight™ Port 1. 0 – TRXIQ mode 1 – JESD207 mode **(default)** **Default**: 01010101 01011001

Address (15 bits)	Bits	Description
0x0024	15 – 14	LML1_S3S[1:0]: Sample source in position 3, when Port 1 is TX. 11 – Sample in frame position 0 is BQ **(default)** 10 – Sample in frame position 0 is BI 01 – Sample in frame position 0 is AQ 00 – Sample in frame position 0 is AI
	13 – 12	LML1_S2S[1:0]: Sample source in position 2, when Port 1 is TX. 11 – Sample in frame position 0 is BQ 10 – Sample in frame position 0 is BI **(default)** 01 – Sample in frame position 0 is AQ 00 – Sample in frame position 0 is AI
	11 – 10	LML1_S1S[1:0]: Sample source in position 1, when Port 1 is TX. 11 – Sample in frame position 0 is BQ 10 – Sample in frame position 0 is BI 01 – Sample in frame position 0 is AQ **(default)** 00 – Sample in frame position 0 is AI
	9 – 8	LML1_S0S[1:0]: Sample source in position 0, when Port 1 is TX. 11 – Sample in frame position 0 is BQ 10 – Sample in frame position 0 is BI 01 – Sample in frame position 0 is AQ 00 – Sample in frame position 0 is AI **(default)**
	7 – 6	LML1_BQP[1:0]: BQ sample position in frame, when Port 1 is RX. 11 – BQ sample position is 3 **(default)** 10 – BQ sample position is 2 01 – BQ sample position is 1 00 – BQ sample position is 0
	5 – 4	LML1_BIP[1:0]: BI sample position in frame, when Port 1 is RX. 11 – BI sample position is 3 10 – BI sample position is 2 **(default)** 01 – BI sample position is 1 00 – BI sample position is 0
	3 – 2	LML1_AQP[1:0]: AQ sample position in frame, when Port 1 is RX. 11 – AQ sample position is 3 10 – AQ sample position is 2 01 – AQ sample position is 1 **(default)** 00 – AQ sample position is 0
	1 – 0	LML1_AIP[1:0]: AI sample position in frame, when Port 1 is RX. 11 – AI sample position is 3 10 – AI sample position is 2 01 – AI sample position is 1 00 – AI sample position is 0 **(default)** **Default**: 11100100 11100100
0x0025	15 – 8	LML1_TX_PST[7:0]: Number of clock cycles to wait before data drive stops after burst stop is detected in JESD207 mode on Port 1 and Port 1 is transmitter. Unsigned integer. Possible values are 0 – 255, **default** is 1.
	7 – 0	LML1_TX_PRE[7:0]: Number of clock cycles to wait before data drive starts after burst start is detected in JESD207 mode on Port 1 and Port 1 is transmitter. Unsigned integer. Possible values are 0 – 255, **default** is 1. **Default**: 00000001 00000001
0x0026	15 – 8	LML1_RX_PST[7:0]: Number of clock cycles to wait before data capture stops after burst stop is detected in JESD207 mode on Port 1 and Port 1 is receiver. Unsigned integer. Possible values are 0 – 255, **default** is 1.
	7 – 0	LML1_RX_PRE[7:0]: Number of clock cycles to wait before data capture starts after burst start is detected in JESD207 mode on Port 1 and Port 1 is receiver. Unsigned integer. Possible values are 0 – 255, **default** is 1. **Default**: 00000001 00000001

Address (15 bits)	Bits	Description

0x0027	15 – 14	LML2_S3S[1:0]: Sample source in position 3, when Port 2 is TX.
		    11 – Sample in frame position 0 is BQ **(default)**
		    10 – Sample in frame position 0 is BI
		    01 – Sample in frame position 0 is AQ
		    00 – Sample in frame position 0 is AI

	13 – 12	LML2_S2S[1:0]: Sample source in position 2, when Port 2 is TX.
		    11 – Sample in frame position 0 is BQ
		    10 – Sample in frame position 0 is BI **(default)**
		    01 – Sample in frame position 0 is AQ
		    00 – Sample in frame position 0 is AI

	11 – 10	LML2_S1S[1:0]: Sample source in position 1, when Port 2 is TX.
		    11 – Sample in frame position 0 is BQ
		    10 – Sample in frame position 0 is BI
		    01 – Sample in frame position 0 is AQ **(default)**
		    00 – Sample in frame position 0 is AI

	9 – 8	LML2_S0S[1:0]: Sample source in position 0, when Port 2 is TX.
		    11 – Sample in frame position 0 is BQ
		    10 – Sample in frame position 0 is BI
		    01 – Sample in frame position 0 is AQ
		    00 – Sample in frame position 0 is AI **(default)**

	7 – 6	LML2_BQP[1:0]: BQ sample position in frame, when Port 2 is RX.
		    11 – BQ sample position is 3 **(default)**
		    10 – BQ sample position is 2
		    01 – BQ sample position is 1
		    00 – BQ sample position is 0

	5 – 4	LML2_BIP[1:0]: BI sample position in frame, when Port 2 is RX.
		    11 – BI sample position is 3
		    10 – BI sample position is 2 **(default)**
		    01 – BI sample position is 1
		    00 – BI sample position is 0

	3 – 2	LML2_AQP[1:0]: AQ sample position in frame, when Port 2 is RX.
		    11 – AQ sample position is 3
		    10 – AQ sample position is 2
		    01 – AQ sample position is 1 **(default)**
		    00 – AQ sample position is 0

	1 – 0	LML2_AIP[1:0]: AI sample position in frame, when Port 2 is RX.
		    11 – AI sample position is 3
		    10 – AI sample position is 2
		    01 – AI sample position is 1
		    00 – AI sample position is 0 **(default)**
		
		Default: 11100100 11100100

0x0028	15 – 8	LML2_TX_PST[7:0]: Number of clock cycles to wait before data drive stops after burst stop is detected in JESD207 mode on Port 2 and Port 2 is transmitter. Unsigned integer. Possible values are 0 – 255, **default** is 1.
	7 – 0	LML2_TX_PRE[7:0]: Number of clock cycles to wait before data drive starts after burst start is detected in JESD207 mode on Port 2 and Port 2 is transmitter. Unsigned integer. Possible values are 0 – 255, **default** is 1. **Default**: 00000001 00000001
0x0029	15 – 8	LML2_RX_PST[7:0]: Number of clock cycles to wait before data capture stops after burst stop is detected in JESD207 mode on Port 2 and Port 2 is receiver. Unsigned integer. Possible values are 0 – 255, **default** is 1.
	7 – 0	LML2_RX_PRE[7:0]: Number of clock cycles to wait before data capture starts after burst start is detected in JESD207 mode on Port 2 and Port 2 is receiver. Unsigned integer. Possible values are 0 – 255, **default** is 1. **Default**: 00000001 00000001

Address (15 bits)	Bits	Description
0x002A	15 – 12	**Reserved**

	11 – 10	RX_MUX[1:0]: RxFIFO data source selection.
		    00 – RxTSPCLK **(default)**
		    01 – TxFIFO
		    10, 11 – LFSR

	9 – 8	TX_MUX[1:0]: Port selection for data transmit to TSP.
		    10, 11 – Data source is RxTSP
		    01 – Data source is Port 2
		    00 – Data source is Port 1 **(default)**

	7 – 6	TXRDCLK_MUX[1:0]: TX FIFO read clock selection.
		    10, 11 – Clock source is TxTSPCLK **(default)**
		    01 – Clock source is FCLK2
		    00 – Clock source is FCLK1

	4 – 5	TXWRCLK_MUX[1:0]: TX FIFO write clock selection.
		    10, 11 – Clock source is RxTSPCLK (use for TSP loop back)
		    01 – Clock source is FCLK2
		    00 – Clock source is FCLK1 **(default)**

	3 – 2	RXRDCLK_MUX[1:0]: RX FIFO read clock selection.
		    11 – Clock source is FCLK2
		    10 – Clock source is FCLK1
		    01 – Clock source is MCLK2 **(default)**
		    00 – Clock source is MCLK1

	1 – 0	RXWRCLK_MUX[1:0]: RX FIFO write clock selection.
		    10, 11 – Clock source is RxTSPCLK **(default)**
		    01 – Clock source is FCLK2
		    00 – Clock source is FCLK1
		
		Default: 00000000 10000110

0x002B	15	FCLK2_INV: FCLK2 clock inversion.
		    1 – Inverted
		    0 – Not inverted **(default)**

	14	FCLK1_INV: FCLK1 clock inversion.
		    1 – Inverted
		    0 – Not inverted **(default)**

	13 – 12	MCLK2DLY[1:0]: MCLK2 clock internal delay.
		    11 – 3x delay
		    10 – 2x delay
		    01 – 1x delay
		    00 – No delay **(default)**

	11 – 10	MCLK1DLY[1:0]: MCLK2 clock internal delay.
		    11 – 3x delay
		    10 – 2x delay
		    01 – 1x delay
		    00 – No delay **(default)**

	9 – 6	**Reserved**

	5 – 4	MCLK2SRC[1:0]: MCLK2 clock source.
		    11 – RxTSPCLKA
		    10 – TxTSPCLKA
		    01 – RxTSPCLKA after divider **(default)**
		    00 – TxTSPCLKA after divider

	3 – 2	MCLK1SRC[1:0]: MCLK1 clock source.
		    11 – RxTSPCLKA
		    10 – TxTSPCLKA
		    01 – RxTSPCLKA after divider
		    00 – TxTSPCLKA after divider **(default)**

	1	TXDIVEN: TX clock divider enable.
		    1 – Divider enabled
		    0 – Divider disabled **(default)**

	0	RXDIVEN: RX clock divider enable.
		    1 – Divider enabled
		    0 – Divider disabled **(default)**
		
		Default: 00000000 00010000

0x002C	15 – 8	TXTSPCLKA_DIV[7:0]: TxTSP clock divider, used to produce MCLK(1/2) clocks. Unsigned integer. Possible values are 0 – 255, **default** is 255.
	7 – 0	RXTSPCLKA_DIV[7:0]: RxTSP clock divider, used to produce MCLK(1/2) clocks. Unsigned integer. Possible values are 0 – 255, **default** is 255. **Default**: 11111111 11111111

Address (15 bits)	Bits	Description
0x002D	15 – 0	**Reserved** **Default**: 11111111 11111111

0x002E	15	MIMO/SISO: MIMO channel B enable control.
		      1 – Disables MIMO channel B, when SISO_ID (from pad) is 1.
		      0 – Enables MIMO channel B, when SISO_ID (from pad) is 0.

	14 – 0	**Reserved** **Default**: 00000000 00000000
0x002F	15 – 7	VER[4:0]: Chip version. **Read only.** 00111 – Chip version is 7
	10 – 6	REV[4:0]: Chip revision. **Read only.** 00001 – Chip revision is 1
	5 – 0	MASK[5:0]: Chip mask. **Read only.** 000000 – Chip mask is 0 **Default**: 00111000 01000000 (**Read only**)

2.3   NCO Configuration Memory

The NCO configuration memory control is listed in this chapter. There are 4 NCOs – two for each transmit and receive MIMO channel.

The carrier frequency f_c generated by NCO could be set using the following formula:

$$f_c = \frac{fcw}{2^{32}} f_{clk}$$

where fcw represents the decimal value of the 32-bit frequency control word and f_{clk} is the NCO clock frequency.

The carrier phase offset can also be adjusted using the 16-bit configuration parameter pho. The carrier phase shift is calculated as follows:

$$\varphi = 2\pi \frac{pho}{2^{16}} \, ,$$

with pho being the decimal value stored in the carrier phase offset register.

Table 3 NCO configuration memory

Address (15 bits)	Bits	Description
TX(A/B): 0x0240 RX(A/B): 0x0440	15 – 9 8 – 5 4 – 1 0	**Reserved** DTHBIT[3:0]: NCO bits to dither. 　　　　0000 – Dithering disabled 　　　　0001 – 1 bit dithering **(default)** 　　　　... 　　　　1111 – 15 bit dithering SEL[3:0]: Selects PHO or FCW to feed to NCO, according to MODE. **Shadow register.** 　　　　0000 – PHO0 or FCW0 selected **(default)** 　　　　0001 – PHO1 or FCW1 selected 　　　　... 　　　　1111 – PHO15 or FCW15 selected MODE: Memory table mode. **Shadow register.** 　　　　1 – PHO table (data at addresses 0x4 to 0x13 are PHO) 　　　　0 – FCW table (data at addresses 0x2 to 0x20 are FCW) **(default)** **Default**: 00000000 00100000
TX(A/B): 0x0241 RX(A/B): 0x0441	15 – 0	PHO[15:0]: NCO Phase offset register, when MODE = 0. **Default**: 00000000 00000000
TX(A/B): 0x0242 RX(A/B): 0x0442	15 – 0	FCW0[31:16]: NCO frequency control word register 0. MSB part. **Default**: 00000000 00000000
TX(A/B): 0x0243 RX(A/B): 0x0443	15 – 0	FCW0[15:0]: NCO frequency control word register 0. LSB part. **Default**: 00000000 00000000
TX(A/B): 0x0244 RX(A/B): 0x0444	15 – 0	FCW1[31:16]: NCO frequency control word register 1, when MODE = 0. MSB part. PHO0[15:0]: NCO Phase offset register 0, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0245 RX(A/B): 0x0445	15 – 0	FCW1[15:0]: NCO frequency control word register 1, when MODE = 0. LSB part. PHO1[15:0]: NCO Phase offset register 1, when MODE = 1. **Default**: 00000000 00000000

Address (15 bits)	Bits	Description
TX(A/B): 0x0246 RX(A/B): 0x0446	15 – 0	FCW2[31:16]: NCO frequency control word register 2, when MODE = 0. MSB part. PHO2[15:0]: NCO Phase offset register 2, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0247 RX(A/B): 0x0447	15 – 0	FCW2[15:0]: NCO frequency control word register 2, when MODE = 0. LSB part. PHO3[15:0]: NCO Phase offset register 3, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0248 RX(A/B): 0x0448	15 – 0	FCW3[31:16]: NCO frequency control word register 3, when MODE = 0. MSB part. PHO4[15:0]: NCO Phase offset register 4, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0249 RX(A/B): 0x0449	15 – 0	FCW3[15:0]: NCO frequency control word register 3, when MODE = 0. LSB part. PHO5[15:0]: NCO Phase offset register 5, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024A RX(A/B): 0x044A	15 – 0	FCW4[31:16]: NCO frequency control word register 4, when MODE = 0. MSB part. PHO6[15:0]: NCO Phase offset register 6, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024B RX(A/B): 0x044B	15 – 0	FCW4[15:0]: NCO frequency control word register 4, when MODE = 0. LSB part. PHO7[15:0]: NCO Phase offset register 7, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024C RX(A/B): 0x044C	15 – 0	FCW5[31:16]: NCO frequency control word register 5, when MODE = 0. MSB part. PHO8[15:0]: NCO Phase offset register 8, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024D RX(A/B): 0x044D	15 – 0	FCW5[15:0]: NCO frequency control word register 5, when MODE = 0. LSB part. PHO9[15:0]: NCO Phase offset register 9, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024E RX(A/B): 0x044E	15 – 0	FCW6[31:16]: NCO frequency control word register 6, when MODE = 0. MSB part. PHO10[15:0]: NCO Phase offset register 10, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x024F RX(A/B): 0x044F	15 – 0	FCW6[15:0]: NCO frequency control word register 6, when MODE = 0. LSB part. PHO11[15:0]: NCO Phase offset register 11, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0250 RX(A/B): 0x0450	15 – 0	FCW7[31:16]: NCO frequency control word register 7, when MODE = 0. MSB part. PHO12[15:0]: NCO Phase offset register 12, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0251 RX(A/B): 0x0451	15 – 0	FCW7[15:0]: NCO frequency control word register 7, when MODE = 0. LSB part. PHO13[15:0]: NCO Phase offset register 13, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0252 RX(A/B): 0x0452	15 – 0	FCW8[31:16]: NCO frequency control word register 8, when MODE = 0. MSB part. PHO14[15:0]: NCO Phase offset register 14, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0253 RX(A/B): 0x0453	15 – 0	FCW8[15:0]: NCO frequency control word register 8, when MODE = 0. LSB part. PHO15[15:0]: NCO Phase offset register 15, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0254 RX(A/B): 0x0454	15 – 0	FCW9[31:16]: NCO frequency control word register 9, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0255 RX(A/B): 0x0455	15 – 0	FCW9[15:0]: NCO frequency control word register 9, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0256 RX(A/B): 0x0456	15 – 0	FCW10[31:16]: NCO frequency control word register 10, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000

Address (15 bits)	Bits	Description
TX(A/B): 0x0257 RX(A/B): 0x0457	15 – 0	FCW10[15:0]: NCO frequency control word register 10, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0258 RX(A/B): 0x0458	15 – 0	FCW11[31:16]: NCO frequency control word register 11, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0259 RX(A/B): 0x0459	15 – 0	FCW11[15:0]: NCO frequency control word register 11, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025A RX(A/B): 0x045A	15 – 0	FCW12[31:16]: NCO frequency control word register 12, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025B RX(A/B): 0x045B	15 – 0	FCW12[15:0]: NCO frequency control word register 12, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025C RX(A/B): 0x045C	15 – 0	FCW13[31:16]: NCO frequency control word register 13, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025D RX(A/B): 0x045D	15 – 0	FCW13[15:0]: NCO frequency control word register 13, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025E RX(A/B): 0x045E	15 – 0	FCW14[31:16]: NCO frequency control word register 14, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x025F RX(A/B): 0x045F	15 – 0	FCW14[15:0]: NCO frequency control word register 14, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0260 RX(A/B): 0x0460	15 – 0	FCW15[31:16]: NCO frequency control word register 15, when MODE = 0. MSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000
TX(A/B): 0x0261 RX(A/B): 0x0461	15 – 0	FCW15[15:0]: NCO frequency control word register 15, when MODE = 0. LSB part. Reserved, when MODE = 1. **Default**: 00000000 00000000

2.4 TxTSP(A/B) Configuration Memory

The block diagrams of TxTSPA and TxTSPB modules are exactly the same. The control structure is shown in *Figure 22*. The tables in this chapter describe the control registers of the TxTSPA and TxTSPB modules.

There is one BIST logic per TxTSPA and TxTSPB. The BIST control structure is shown in Figure 25.

Table 4 TxTSP configuration memory

Address (15 bits)	Bits	Description
0x0200	15 – 10	**Reserved**
	9	TSGFC: TSG full scale control.
		0 – -6dB (**default**)
		1 – Full scale
	8 – 7	TSGFCW: Set frequency of TSG's NCO.
		DC TSG NCO frequency
		=========================
		00 do not use
		01 TSP clk/8 (**default**)
		10 TSP clk/4
		11 do not use
	6	TSGDCLDQ: Load TSG DC Q register with value from DC_REG[15:0].
		0 – No action (**default**)
		0-to-1 – positive edge loads TSG's DC register Q.
	5	TSGDCLDI: Load TSG DC I register with value from DC_REG[15:0].
		0 – No action (**default**)
		0-to-1 – positive edge loads TSG's DC register I.
	4	TSGSWAPIQ: Swap signals at test signal generator's output.
		0 – Do not swap **(default)**
		1 – Swap I an Q signal sources comming from TSG
	3	TSGMODE: Test signal generator mode.
		0 – NCO **(default)**
		1 – DC source
	2	INSEL: Input source of TxTSP:
		0 – LML output **(default)**
		1 – Test signal generator
	1	BSTART: Starts TxTSP built in self test. Keep it at 1 one at least three clock cycles.
		0 – **(default)**
		0-to-1 – positive edge activates BIST
	0	EN: TxTSP modules enable.
		0 – Disabled
		1 – Enabled **(default)**
		Default: 00000000 10000001
0x0201	15 – 11	**Reserved**
	10 – 0	GCORRQ[10:0]: Gain corrector value, channel Q. Unsigned integer.
		Possible values are 0 – 2047, **default** is 2047
		Default: 00000111 11111111
0x0202	15 – 11	**Reserved**
	10 – 0	GCORRI[10:0]:Gain corrector value, channel I Unsigned integer.
		Possible values are 0 to 2047, **default** is 2047
		Default: 00000111 11111111
0x0203	15	**Reserved**
	14 – 12	HBI_OVR[2:0]: HBI interpolation ratio. Interpolation ratio is 2^{HBI_OVR+1}.
		000 – Interpolation ratio is 2 **(default)**
		001 – Interpolation ratio is 4
		010 – Interpolation ratio is 8
		011 – Interpolation ratio is 16
		100 – Interpolation ratio is 32
		111 – Bypass
	11 – 0	IQCORR[11:0]: Phase corrector value (tan(Alpha/2)). Integer, 2's complement.
		Possible values are -2048 to 2047, **default** is 0
		Default: 00000000 00000000

Address (15 bits)	Bits	Description
0x0204	15 – 8	DCCORRI[7:0]: DC corrector value, channel I. Integer, 2's complement. Possible values are -128 to 127, **default** is 0
	7 – 0	DCCORRQ[7:0]: DC corrector value, channel Q. Integer, 2's complement. Possible values are -128 to 127, **default** is 0 **Default**: 00000000 00000000
0x0205	15 – 11	**Reserved**
	10 – 8	GFIR1_L[2:0]: Parameter l of GFIR1 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR1_N[7:0]: Clock division ratio of GFIR1 is GFIR1_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0206	15 – 11	**Reserved**
	10 – 8	GFIR2_L[2:0]: Parameter l of GFIR2 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR2_N[7:0]: Clock division ratio of GFIR2 is GFIR2_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0207	15 – 11	**Reserved**
	10 – 8	GFIR3_L[2:0]: Parameter l of GFIR3 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR3_N[7:0]: Clock division ratio of GFIR3 is GFIR3_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0208	15 – 14	CMIX_GAIN: Gain of CMIX output. 00 – CMIX output gain is 0dB **(default)** 01 – CMIX output gain is +6dB 10, 11 – CMIX output gain is –6dB
	13	CMIX_SC: Spectrum control of CMIX. 1 – Downconvert 0 – Upconvert **(default)**
	12 – 9	**Reserved**
	8	CMIX_BYP: CMIX bypass. 1 – Bypass 0 – Use **(default)**
	7	ISINC_BYP: ISINC bypass. 1 – Bypass 0 – Use **(default)**
	6	GFIR3_BYP: GFIR3 bypass. 1 – Bypass 0 – Use **(default)**
	5	GFIR2_BYP: GFIR2 bypass. 1 – Bypass 0 – Use **(default)**
	4	GFIR1_BYP: GFIR1 bypass. 1 – Bypass 0 – Use **(default)**
	3	DC_BYP: DC corrector bypass. 1 – Bypass 0 – Use **(default)**
	2	**Reserved**
	1	GC_BYP: Gain corrector bypass. 1 – Bypass 0 – Use **(default)**
	0	PH_BYP: Phase corrector bypass. 1 – Bypass 0 – Use **(default)** **Default**: 00000000 00000000
0x0209	15 – 1	BSIGI[14:0]: TxTSP BIST signature, channel I, LSB.
	0	BSTATE: TxTSP BIST state indicator 0 – BIST is not running 1 – BIST in progress **Read only**
0x020A	15 – 8	BSIGQ[7:0]: TxTSP BIST signature, channel Q, LSB.
	7 – 0	BSIGI [22:15]: TxTSP BIST signature, channel I, MSB. **Read only**

Address (15 bits)	Bits	Description
0x020B	15 14 – 0	**Reserved** BSIGQ[22:8]: TxTSP BIST signature, channel Q, MSB. **Read only**
0x020C	15 – 0	DC_REG[15:0]: DC data source for test purposes. Possible values: 2^16-1 – 0 **(default)** **Default**: 00000000 00000000

2.5 RxTSP(A/B) Configuration Memory

The block diagrams of the RxTSPA and RxTSPB modules are exactly the same. The control structure is shown in *Figure 23*. The tables in this chapter describe the control registers of the RxTSPA and RxTSPB modules.

There is one BIST logic per RxTSPA and RxTSPB. The BIST control structure is shown in Figure 26.

Table 5 RxTSP configuration memory

Address (15 bits)	Bits	Description
0x0400	15	CAPTURE: Captures value, selected by CAPSEL[1:0]. 　　　0 – **(default)** 　　　0-to-1 – positive edge captures value, selected by CAPSEL[1:0]
	14 – 13	CAPSEL[1:0]: Selects what parameters to capture to memory (addresses 0x0400E and 0x0400F) 　　　00 – RSSI **(default)** 　　　01 – ADCI and ADCQ 　　　10 – BSIGI and BSTATE 　　　11 – BSIGQ and BSTATE
	12 – 10	**Reserved**
	9	TSGFC: TSG full scale control. 　　　0 – -6dB **(default)** 　　　1 – Full scale
	8 – 7	TSGFCW[1:0]: Set frequency of TSG's NCO. 　　　DC　　　TSG NCO frequency 　　　====================== 　　　00　　　do not use 　　　01　　　TSP clk/8 **(default)** 　　　10　　　TSP clk/4 　　　11　　　do not use
	6	TSGDCLDQ: Load TSG DC Q register with value from DC_REG[15:0]. 　　　0 – No action **(default)** 　　　0-to-1 – positive edge loads TSG's DC register Q.
	5	TSGDCLDI: Load TSG DC I register with value from DC_REG[15:0]. 　　　0 – No action **(default)** 　　　0-to-1 – positive edge loads TSG's DC register I.
	4	TSGSWAPIQ: Swap signals at test signal generator's output. 　　　0 – Do not swap **(default)** 　　　1 – Swap I an Q signal sources comming from TSG
	3	TSGMODE: Test signal generator mode. 　　　0 – NCO **(default)** 　　　1 – DC source
	2	INSEL: Input source of TxTSP: 　　　0 – LML output **(default)** 　　　1 – Test signal generator
	1	BSTART: Starts delta sigma built in self test. Keep it at 1 one at least three clock cycles. 　　　0 – **(default)** 　　　0-to-1 – positive edge activates BIST
	0	EN: RxTSP modules enable. 　　　0 – Disabled 　　　1 – Enabled **(default)** **Default**: 00000000 10000001
0x0401	15 – 11	**Reserved**
	10 – 0	GCORRQ[10:0]: Gain corrector value, channel Q. Unsigned integer. 　　　Possible values are 0 – 2047, **default** is 2047 **Default**: 00000111 11111111
0x0402	15 – 11	**Reserved**
	10 – 0	GCORRI[10:0]:Gain corrector value, channel I Unsigned integer. 　　　Possible values are 0 to 2047, **default** is 2047 **Default**: 00000111 11111111

Address (15 bits)	Bits	Description
0x0403	15	**Reserved**
	14 – 12	HBD_OVR[2:0]: HBD decimation ratio. Decimation ratio is 2^{HBD_OVR+1}. 000 – Decimation ratio is 2 **(default)** 001 – Decimation ratio is 4 010 – Decimation ratio is 8 011 – Decimation ratio is 16 100 – Decimation ratio is 32 111 – Bypass
	11 – 0	IQCORR[11:0]: Phase corrector value (tan(Alpha/2)). Integer, 2's complement. Possible values are -2048 to 2047, **default** is 0 **Default**: 00000000 00000000
0x0404	15 – 3	**Reserved**
	2 – 0	DCCORR_AVG[2:0]: Number of samples to average for Automatic DC corrector. Number of samples to average is $2^{DCCORR\_AVG + 12}$. **Default**: 00000000 00000000
0x0405	15 – 11	**Reserved**
	10 – 8	GFIR1_L[2:0]: Parameter l of GFIR1 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR1_N[7:0]: Clock division ratio of GFIR1 is GFIR1_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0406	15 – 11	**Reserved**
	10 – 8	GFIR2_L[2:0]: Parameter l of GFIR2 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR2_N[7:0]: Clock division ratio of GFIR2 is GFIR2_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0407	15 – 11	**Reserved**
	10 – 8	GFIR3_L[2:0]: Parameter l of GFIR3 (l = roundUp(CoeffN/5)-1). Unsigned integer. Possible values are 0 to 7, **default** is 0
	7 – 0	GFIR3_N[7:0]: Clock division ratio of GFIR3 is GFIR3_N + 1. Unsigned integer. Possible values are 0 to 255, **default** is 0 **Default**: 00000000 00000000
0x0408	15 – 0	AGC_K[15:0]: AGC loop gain, LSB. **Default**: 00000000 00000000
0x0409	15 – 4	AGC_ADESIRED[11:0]: Desired output signal level.
	3 – 2	**Reserved**
	1 – 0	AGC_K[17:16]: AGC loop gain, MSB. **Default**: 00000000 00000000
0x040A	15 – 14	**Reserved**
	13 – 12	AGC_MODE[1:0]: AGC Mode. 0 – AGC mode 1 – RSSI mode 2, 3 – Bypass
	11 – 3	**Reserved**
	2 – 0	AGC_AVG[2:0]: AGC averaging window size is $2^{(AGC\_AVG + 7)}$. **Default**: 00000000 00000000
0x040B	15 – 0	DC_REG[15:0]: DC data source for test purposes. Possible values: 2^16-1 – 0 **(default)** **Default**: 00000000 00000000

Address (15 bits)	Bits	Description
0x040C	15 – 14	CMIX_GAIN: Gain of CMIX output. 00 – CMIX output gain is 0dB **(default)** 01 – CMIX output gain is +6dB 10, 11 – CMIX output gain is –6dB
	13	CMIX_SC: Spectrum control of CMIX. 1 – Downconvert 0 – Upconvert **(default)**
	12 – 8	**Reserved**
	7	CMIX_BYP: CMIX bypass. 1 – Bypass 0 – Use **(default)**
	6	AGC_BYP: AGC bypass. 1 – Bypass 0 – Use **(default)**
	5	GFIR3_BYP: GFIR3 bypass. 1 – Bypass 0 – Use **(default)**
	4	GFIR2_BYP: GFIR2 bypass. 1 – Bypass 0 – Use **(default)**
	3	GFIR1_BYP: GFIR1 bypass. 1 – Bypass 0 – Use **(default)**
	2	DC_BYP: DC corrector bypass. 1 – Bypass 0 – Use **(default)**
	1	GC_BYP: Gain corrector bypass. 1 – Bypass 0 – Use **(default)**
	0	PH_BYP: Phase corrector bypass. 1 – Bypass 0 – Use **(default)** **Default**: 00000000 00000000
0x040E and 0x040F	15 – 8 7 – 0	CAPD[31:0]: Data capture register. Stores data, selected by CAPSEL[1:0], on rising edge of CAPTURE. Register layout is as follows: CAPSEL[1:0] 0x040E 0x040F === 00 00000000000000, RSSI[1:0] RSSI[17:2] 01 000000, ADCI[9:0] 000000, ADCQ[9:0] 10 BISTI[14:0], BSTATE 00000000, BISTI[22:15] 11 BISTQ[14:0], BSTATE 00000000, BISTQ[22:15] **Read only**

2.6 RX/TX GFIR1/GFIR2 Coefficient Memory

The general purpose digital FIR filter (GFIR1 and GFIR2) coefficients are stored in the following tables.

Table 6 Memory space used to store TxGFIR1/RxGFIR1 coefficients

Address (15 bits)	Bits	Description
Tx: 0x0280 – 0x0287 Rx: 0x0480 – 0x0487	8 x 16	Tx(Rx)1CMB0[7:0][15:0]: Coefficients memory bank 0 for TxGFIR1/RxGFIR1.
Tx: 0x0288 – 0x028F Rx: 0x0488 – 0x048F	8 x 16	Tx(Rx)1CMB1[7:0][15:0]: Coefficients memory bank 1 for TxGFIR1/RxGFIR1.
Tx: 0x0290 – 0x0297 Rx: 0x0490 – 0x0497	8 x 16	Tx(Rx)1CMB2[7:0][15:0]: Coefficients memory bank 2 for TxGFIR1/RxGFIR1.
Tx: 0x0298 – 0x029F Rx: 0x0498 – 0x049F	8 x 16	Tx(Rx)1CMB3[7:0][15:0]: Coefficients memory bank 3 for TxGFIR1/RxGFIR1.
Tx: 0x02A0 – 0x02A7 Rx: 0x04A0 – 0x04A7	8 x 16	Tx(Rx)1CMB4[7:0][15:0]: Coefficients memory bank 4 for TxGFIR1/RxGFIR1.
Tx: 0x02A8 – 0x02BF Rx: 0x04A8 – 0x04BF	24 x 16	Reserved

Table 7 Memory space used to store TxGFIR2/RxGFIR2 coefficients

Address (15 bits)	Bits	Description
Tx: 0x02C0 – 0x02C7 Rx: 0x04C0 – 0x04C7	8 x 16	Tx(Rx)2CMB0[7:0][15:0]: Coefficients memory bank 0 for TxGFIR2/RxGFIR2.
Tx: 0x02C8 – 0x02CF Rx: 0x04C8 – 0x04CF	8 x 16	Tx(Rx)2CMB1[7:0][15:0]: Coefficients memory bank 1 for TxGFIR2/RxGFIR2.
Tx: 0x02D0 – 0x02D7 Rx: 0x04D0 – 0x04D7	8 x 16	Tx(Rx)2CMB2[7:0][15:0]: Coefficients memory bank 2 for TxGFIR2/RxGFIR2.
Tx: 0x02D8 – 0x02DF Rx: 0x04D8 – 0x04DF	8 x 16	Tx(Rx)2CMB3[7:0][15:0]: Coefficients memory bank 3 for TxGFIR2/RxGFIR2.
Tx: 0x02E0 – 0x02E7 Rx: 0x04E0 – 0x04E7	8 x 16	Tx(Rx)2CMB4[7:0][15:0]: Coefficients memory bank 4 for TxGFIR2/RxGFIR2.
Tx: 0x02E8 – 0x02FF Rx: 0x04E8 – 0x04FF	24 x 16	Reserved

2.7 RX/TX GFIR3 Coefficient Memory

The general purpose digital FIR filter (GFIR3) coefficients are stored in the following table.

Table 8 Memory space used to store TxGFIR3 coefficients

Address (15 bits)	Bits	Description
Tx: 0x0300 – 0x0307 Rx: 0x0500 – 0x0507	8 x 16	Tx(Rx)3CMB0a[7:0][15:0]: Coefficients memory bank 0a for TxGFIR2/RxGFIR3.
Tx: 0x0308 – 0x030F Rx: 0x0508 – 0x050F	8 x 16	Tx(Rx)3CMB1a[7:0][15:0]: Coefficients memory bank 1a for TxGFIR2/RxGFIR3.
Tx: 0x0310 – 0x0317 Rx: 0x0510 – 0x0517	8 x 16	Tx(Rx)3CMB2a[7:0][15:0]: Coefficients memory bank 2a for TxGFIR2/RxGFIR3.
Tx: 0x0318 – 0x031F Rx: 0x0518 – 0x051F	8 x 16	Tx(Rx)3CMB3a[7:0][15:0]: Coefficients memory bank 3a for TxGFIR2/RxGFIR3.
Tx: 0x0320 – 0x0327 Rx: 0x0520 – 0x0527	8 x 16	Tx(Rx)3CMB4a[7:0][15:0]: Coefficients memory bank 4a for TxGFIR2/RxGFIR3.
Tx: 0x0328 – 0x033F Rx: 0x0528 – 0x053F	24 x 16	Reserved
Tx: 0x0340 – 0x0347 Rx: 0x0540 – 0x0547	8 x 16	Tx(Rx)3CMB0b[7:0][15:0]: Coefficients memory bank 0b for TxGFIR2/RxGFIR3.
Tx: 0x0348 – 0x034F Rx: 0x0548 – 0x054F	8 x 16	Tx(Rx)3CMB1b[7:0][15:0]: Coefficients memory bank 1b for TxGFIR2/RxGFIR3.
Tx: 0x0350 – 0x0357 Rx: 0x0550 – 0x0557	8 x 16	Tx(Rx)3CMB2b[7:0][15:0]: Coefficients memory bank 2b for TxGFIR2/RxGFIR3.
Tx: 0x0358 – 0x035F Rx: 0x0558 – 0x055F	8 x 16	Tx(Rx)3CMB3b[7:0][15:0]: Coefficients memory bank 3b for TxGFIR2/RxGFIR3.
Tx: 0x0360 – 0x0367 Rx: 0x0560 – 0x0567	8 x 16	Tx(Rx)3CMB4b[7:0][15:0]: Coefficients memory bank 4b for TxGFIR2/RxGFIR3.
Tx: 0x0368 – 0x037F Rx: 0x0568 – 0x057F	24 x 16	Reserved
Tx: 0x0380 – 0x0387 Rx: 0x0580 – 0x0587	8 x 16	Tx(Rx)3CMB0c[7:0][15:0]: Coefficients memory bank 0c for TxGFIR2/RxGFIR3.
Tx: 0x0388 – 0x038F Rx: 0x0588 – 0x058F	8 x 16	Tx(Rx)3CMB1c[7:0][15:0]: Coefficients memory bank 1c for TxGFIR2/RxGFIR3.
Tx: 0x0390 – 0x0397 Rx: 0x0590 – 0x0597	8 x 16	Tx(Rx)3CMB2c[7:0][15:0]: Coefficients memory bank 2c for TxGFIR2/RxGFIR3.
Tx: 0x0398 – 0x039F Rx: 0x0598 – 0x059F	8 x 16	Tx(Rx)3CMB3c[7:0][15:0]: Coefficients memory bank 3c for TxGFIR2/RxGFIR3.
Tx: 0x03A0 – 0x03A7 Rx: 0x05A0 – 0x05A7	8 x 16	Tx(Rx)3CMB4c[7:0][15:0]: Coefficients memory bank 4c for TxGFIR2/RxGFIR3.
Tx: 0x03A8 – 0x03BF Rx: 0x05A8 – 0x05BF	24 x 16	Reserved

2.8 RFE(1, 2) Configuration Memory

The block diagrams of the RFE1 and RFE2 modules are shown in *Figure 5* and *Figure 6* respectively. The tables in this chapter describes control registers of RFE1 and RFE2 modules.

Table 9: RFE(1, 2) configuration memory

Address (15 bits)	Bits	Description
0x010C	15 – 12	CDC_I_RFE_(1,2)[3:0]: Trims the duty cycle in I channel. **Default = 8;**
	11 – 8	CDC_Q_RFE_(1,2)[3:0]: Trims the duty cycle in Q channel. **Default = 8;**

	7	PD_LNA_RFE_(1, 2): Power control signal for LNA_RFE
		    0 – block active
		    1 – block powered down **(default)**

	6	PD_RLOOPB_1_RFE_(1, 2): Power control signal for RXFE loopback 1
		    0 – block active
		    1 – block powered down **(default)**

	5	PD_RLOOPB_2_RFE_(1, 2): Power control signal for RXFE loopback 2
		    0 – block active
		    1 – block powered down **(default)**

	4	PD_MXLOBUF_RFE_(1, 2): Power control signal for RXFE mixer lo buffer
		    0 – block active
		    1 – block powered down **(default)**

	3	PD_QGEN_RFE_(1, 2): Power control signal for RXFE quadrature LO generator
		    0 – block active
		    1 – block powered down **(default)**

	2	PD_RSSI_RFE_(1, 2): Power control signal for RXFE RSSI
		    0 – block active
		    1 – block powered down **(default)**

	1	PD_TIA_RFE_(1, 2): Power control signal for RXFE TIA
		    0 – block active **(default)**
		    1 – block powered down

	0	EN_G_RFE_(1, 2): Enable control for all the RFE_1 power downs
		    0 – All RFE_1 modules powered down
		    1 – All RFE_1 modules controlled by individual power down registers **(default)**
		
		Default: 10001000 11111101

Address (15 bits)	Bits	Description
0x010D	15 – 9	**Reserved**

	8 – 7	SEL_PATH_RFE_(1, 2): Selects the active path of the RXFE
		    0 – No path active
		    1 – LNAH path active **(default)**
		    2 – LNAL path active
		    3 – LNAW path active

	6	EN_DCOFF_RXFE_RFE_(1, 2): Enables the DCOFFSET block for the RXFE
		    0 – disabled **(default)**
		    1 – enabled

	5	**Reserved**

	4	EN_INSHSW_LB1_RFE_(1, 2): Enables the input shorting switch at the input of the loopback 1 (in parallel with LNAL mixer). Switch ON resistance < 3ohm
		    0 – switch OFF
		    1 – switch ON **(default)**
		    Should be '1' when RXFE loopback1 is NOT active

	3	EN_INSHSW_LB2_RFE_(1, 2): Enables the input shorting switch at the input of the loopback 2 (in parallel with LNAW mixer)
		Switch ON resistance < 3ohm
		    0 – switch OFF
		    1 – switch ON **(default)**
		    Should be '1' when RXFE Loopback2 is NOT active

	2	EN_INSHSW_L_RFE_(1, 2): Enables the input shorting switch at the input of the LNAL
		Switch ON resistance < 3ohm
		    0 – switch OFF
		    1 – switch ON **(default)**
		    Should be '1' when LNAL is NOT active

	1	EN_INSHSW_W_RFE_(1, 2): Enables the input shorting switch at the input of the LNAW. Switch ON resistance < 3ohm
		    0 – switch OFF
		    1 – switch ON **(default)**
		    Should be '1' when LNAW is NOT active

	0	EN_NEXTRX_RFE_(1, 2): Enables the daisy chain LO buffer going from RXFE1 to RXFE2.
		    0 – SISO **(default)**
		    1 – MIMO
		
		Default: 00000000 10011110

0x010E	15 – 14	**Reserved**

	13 – 7	DCOFFI_RFE_(1, 2)[6:0]: Controls DC offset at the output of the TIA by injecting current t o the input of the TIA. (For I side). **Default: 64**
		    DCOFFSETx_RFE[6] – sign.
		    DCOFFSETx_RFE[5:0] – magnitude. When DCOFFSETx_RFE[5:0] 0,
		    0 current is inejction – no added noise.

	6 – 0	DCOFFQ_RFE_(1, 2)[6:0]: Controls DC offset at the output of the TIA by injecting current t o the input of the TIA. (For Q side). **Default: 64**
		    DCOFFSETx_RFE[6] – sign.
		    DCOFFSETx_RFE[5:0] – magnitude. When DCOFFSETx_RFE[5:0] 0,
		    0 current is inejction – no added noise.
		
		Default: 00100000 01000000

0x010F	15	**Reserved**
	14 – 10	ICT_LOOPB_RFE_(1, 2)[4:0]: Controls the reference current of the RXFE loopback amplifier. **Default: 12** I supply = I supply nominal *(ICT/12).
	9 – 5	ICT_TIAMAIN_RFE_(1, 2)[4:0]: Controls the reference current of the RXFE TIA first stage. **Default: 12** I supply = I supply nominal *(ICT/12).
	4 – 0	ICT_TIAOUT_RFE_(1, 2)[4:0]: Controls the reference current of the RXFE TIA 2nd stage. **Default: 12** I supply = I supply nominal *(ICT/12). **Default: 00110001 10001100**

Address (15 bits)	Bits	Description

0x0110	15
	
	14 – 10
	
	9 – 5
	
	4 – 0
		Reserved
		
		ICT_LNACMO_RFE_(1, 2)[4:0]: Controls the current generating LNA output common mode voltage. **Default: 2**
		
		ICT_LNA_RFE_(1, 2)[4:0]: Controls the current of the LNA core. **Default: 12**
		    Block current = Nominal current * (ICT / 12)
		
		ICT_LODC_RFE_(1, 2)[4:0]: Controls the DC of the mixer LO signal at the gate of the mixer switches. **Default: 20**
		    Vgate=Vth+3.5 kohm*20uA*(ICT/12)
		    If Vgate is too high, the voltage saturates and further increasing this ICT will not increase Vgate. Possible over voltage on mixer gates.
		
		Default: 00001001 10010100

0x0111	15 – 10
	9 – 5
	
	4 – 0
		Reserved
		CAP_RXMXO_RFE_(1, 2)[4:0]: Control the decoupling cap at the output of the RX Mixer. **Default: 4**
		    SE cap = (CAP_RXMXO_RFE +1) * 80fF
		CGSIN_LNA_RFE_(1, 2)[4:0]: Controls the cap parallel with the LNA input NMOS CGS to control the Q of the matching circuit and provides trade off between gain/NF and IIP. The higher the frequency, the lower CGSIN_LNA_RFE should be. Also, the higher CGSIN, the lower the Q, The lower the gain, the higher the NF, and the higher the IIP3
		    0 – for 3500MHz
		    1 – for 2600MHz
		    3 – for 1900MHz **(default)**
		    6 – for 800MHz
		
		Default: 00000000 10000011

0x0112	15 – 12
	
	11 – 0
		CCOMP_TIA_RFE_(1, 2)[3:0]: Compensation capacitor for TIA. This is a function of CFB_TIA_RFE
		    3 – for CFB 220
		    11 – for CFB 500
		    12 – for CFB>1000 **(default)**
		CFB_TIA_RFE_(1, 2)[11:0]: Feedback capacitor for TIA. Controls the 3 dB BW of the TIA. Should be set with calibration through the digital base band. **Default: 230**
		Nominal values:
		    232 – for F3dB 9.15M
		    468 – for F3dB 4.59M
		
		Default: 11000000 11100110

Address (15 bits)	Bits	Description
0x0113	15 – 10	**Reserved**
	9 – 6	G_LNA_RFE_(1, 2)[3:0]: Controls the gain of the LNA
		15 – Gmax **(default)**
		14 – Gmax-1
		13 – Gmax-2
		12 – Gmax-3
		11 – Gmax-4
		10 – Gmax-5
		9 – Gmax-6
		8 – Gmax-9
		7 – Gmax-12
		6 – Gmax-15
		5 – Gmax-18
		4 – Gmax-21
		3 – Gmax-24
		2 – Gmax-27
		1 – Gmax-30
	5 – 2	G_RXLOOPB_RFE_(1, 2)[3:0]: Controls RXFE loopback gain
		15 – Gmax
		14 – Gmax-0.5
		13 – Gmax-1
		12 – Gmax-1.6
		11 – Gmax-2.4
		10 – Gmax-3
		9 – Gmax-4
		8 – Gmax-5
		7 – Gmax-6.2
		6 – Gmax-7.5
		5 – Gmax-9
		4 – Gmax-11
		3 – Gmax-14
		2 – Gmax-17
		1 – Gmax-24
		0 – Gmax-40 **(default)**
		Should be '0' when actual LNAs are working
	1 – 0	G_TIA_RFE_(1, 2)[1:0]: Controls the Gain of the TIA.
		3 – Gmax **(default)**
		2 – Gmax-3
		1 – Gmax-12
		0 - Not allowed
		Default: 00000011 11000011
0x0114	15 – 9	**Reserved**
	8 – 5	RCOMP_TIA_RFE_(1, 2)[3:0]: Controls the compensation resistors of the TIA operational amplifier.
		11 – for CFB = 220
		4 – **(default)**
	4 – 0	5 – for CFB > 500
		RFB_TIA_RFE_(1, 2)[4:0]: Sets the feedback resistor to the nominal value. This is set using the rppolywo calibration code from the bias block (BIAS_RPPOLY_calibration). **Default: 13**
		Default: 00000000 10001101

2.9 RBB(1, 2) Configuration Memory

The block diagrams of RBB1 and RBB2 modules are shown in *Figure 7* and *Figure 8* respectively. The tables in this chapter describe the control registers of RBB1 and RBB2 modules.

Table 10: RBB(1, 2) configuration memory

Address (15 bits)	Bits	Description

0x0115	15	EN_LB_LPFH_RBB_(1, 2): This is the loopback enable signal that is enabled when high band LPFH_RBB is selected for the loopback path that connects the loopb_lpfi inputs to the virtual ground of the LPFH_RBB block.
		    1 – enabled
		    0 – disabled **(default)**
		    Note: Only one of EN_LB_LPFH_RBB/EN_LB_LPFL_RBB can be enabled concurrently.

	14	EN_LB_LPFL_RBB_(1, 2): This is the loopback enable signal that is enabled when the high-band low pass filter LPFL_RBB is selected for the loopback path that connects the loopb_lpf inputs to the virtual ground of the LPFL_RBB block.
		    1 – enabled
		    0 – disabled **(default)**
		    Note: Only one of EN_LB_LPFH_RBB/EN_LB_LPFL_RBB can be enabled concurrently.

	13 – 4	**Reserved**

	3	PD_LPFH_RBB_(1, 2): Power down of the LPFH block.
		    0 – active
		    1 – powered down **(default)**

	2	PD_LPFL_RBB_(1, 2): Power down of the LPFL block.
		    0 – active **(default)**
		    1 – powered down

	1	PD_PGA_RBB_(1, 2): Power down of the PGA block.
		    0 – active **(default)**
		    1 – powered down

	0	EN_G_RBB_(1, 2): Enable control for all the RBB_1 power downs
		    0 – All RBB modules powered down
		    1 – All RBB modules controlled by individual power down registers **(default)**
		
		Default: 00000000 00001001

0x0116	15 – 11	R_CTL_LPF_RBB_(1, 2)[4:0]: Controls the absolute value of the resistance of the RC time constant of the RBB_LPF blocks (both Low and High). This value is corrected during the calibration process. **Default: 16**

	10 – 8	RCC_CTL_LPFH_RBB_(1, 2)[2:0]: Controls the stability passive compensation of the LPFH_RBB operational amplifier. **Default: 1**
		    1 – when rxMode is 37 MHz,
		    4 – when rxMode 66 MHz,
		    7 – when rxMode 108 MHz

	7 – 0	C_CTL_LPFH_RBB_(1, 2)[7:0]: Controls the capacitance value of the RC time constant of RBB_LPFH and it varies with the respective rxMode from 37 MHz to 108 MHz. Its value is equal to (120*108M/rxMode)*ccor-cfrH; where: rxMode is the receiver mode of operation 37 MHz up to 108 MHz, ccor is determined by calibration and cfrL is valued at 56. This control signal can be determined by lookup tables generated during the calibration phase. **Default: 128** **Default**: 10000001 00000000

Address (15 bits)	Bits	Description
0x0117	15 – 14	**Reserved**
	13 – 11	RCC_CTL_LPFL_RBB_(1, 2)[2:0]: Controls the stability passive compensation of the LPFL_RBB operational amplifier. 　　　　0 – when rxMode is 1.4MHz, 　　　　1 – when 3MHz 　　　　2 – when 5MHz 　　　　3 – when 10MHz 　　　　4 – when 15MHz 　　　　5 – when 20MHz **(default)**
	10 – 0	C_CTL_LPFL_RBB_(1, 2)[10:0]: Controls the capacitance value of the RC time constant of RBB_LPFL and it varies with the respective rxMode from 1.4 MHz to 20 MHz. Its value is equal to (120*20M/rxMode)*ccor-cfrL ; where: rxMode is the receiver mode of operation from 1.4 MHz up to 20 MHz, ccor is determined by calibration and cfrL is valued at 100. This control signal can be determined by lookup tables generated during the calibration phase. **Default: 12** **Default**: 00101000 00001100
0x0118	15-13	INPUT_CTL_PGA_RBB_(1, 2)[2:0]: There are a total of four different differential inputs to the PGA. Only one of them is active at a time. 　　　　0 – when LPFL input is selected (rxMode [=20 MHz]; The output of the LPFL_RBB block is selected as input. **(default)** 　　　　1 – when LPFH input is selected (rxMode] 20 MHz); The output of the LPFH_RBB is selected as input. 　　　　2 – when bypassing the LPF blocks; The input signal to either RBB_LPFH or RBB_LPFL is bypassed and connected directly to the PGA bypass input. 　　　　3 – when connecting loopb_tx (the loop back from TBB) to the input of the PGA. 　　　　4 – when loopb_pkd (Loop back path from the peak detector) is selected.
	12 – 10	**Reserved**
	9 – 5	ICT_LPF_IN_RBB_(1, 2)[4:0]: Controls the reference bias current of the input stage of the operational amplifier used in RBB_LPF blocks (Low or High). Must increase up to 24 when a strong close blocker is detected to maintain the linearity performance. **Default: 12**
	4 – 0	ICT_LPF_OUT_RBB_(1, 2)[4:0]: Controls the reference bias current of the output stage of the operational amplifier used in RBB_LPF blocks (low or High). Must increase up to 24 when a strong close blocker is detected to maintain the linearity performance. **Default: 12** **Default**: 00000001 10001100
0x0119	15	OSW_PGA_RBB_(1, 2): There are two instances of the PGA circuit in the design. The outputs of the RBB_LPF blocks are connected to the input of these PGA blocks (common). The output of one of them is connected to two pads pgaoutn and pgaoutp and the output of the other PGA is connected directly to the ADC input. 　　　　0 – the PGA connected to the ADC is selected; **(default)** 　　　　1 – the PGA connected to the output pads is selected instead.
	14 – 10	ICT_PGA_OUT_RBB_(1, 2)[4:0]: Controls the output stage reference bias current of the operational amplifier used in the PGA circuit. Must increase up to 12 when a strong close blocker is detected or when operating at the high band frequencies to maintain the linearity performance. **Default: 6**
	9 – 5	ICT_PGA_IN_RBB_(1, 2)[4:0]: Controls the input stage reference bias current of the operational amplifier used in the PGA circuit. Must increase up to 12 when a strong close blocker is detected or when operating at the high band frequencies to maintain the linearity performance. **Default: 6**
	4 – 0	G_PGA_RBB_(1, 2)[4:0]: This is the gain of the PGA. The gain is adaptively set to maintain a signal swing of 0.6 Vpkd at the output of the PGA. The value of the gain is: Gain(dB) = -12+G_PGA_RBB. **Default: 11** **Default**: 00011000 11001011

Address (15 bits)	Bits	Description
0x011A	15 – 14	**Reserved**
	13 – 9	RCC_CTL_PGA_RBB_(1, 2)[4:0]: Controls the stability passive compensation of the PGA_RBB operational amplifier. Its value is equal to: (430f*(0.65**(G_PGA_RBB/10))-110.35f)/20.4516f + 16 when ICT_PGA is 12. An offline/off chip lookup table can be generated and stored. **Default: 23**
	8	**Reserved**

	7 – 0	C_CTL_PGA_RBB_(1, 2)[7:0]: Controls the value of the feedback capacitor of the PGA that is used to help against the parasitic cap at the virtual node for stability.
		       3 – when 0<=G_PGA_RBB<8
		       2 – when 8<=G_PGA_RBB<13 **(default)**
		       1 – when 13<=G_PGA_RBB<21
		       0 – when 21<=G_PGA_RBB
		
		Default: 00101110 00000010

0x011B	15 – 7	**Reserved**
	6 – 0	RESRV_RBB_(1, 2)[6:0]: Reserved for future use. **Default: 0** **Default**: 00000000 00000000

2.10 TRF(1, 2) Configuration Memory

The block diagrams of the TRF1 and TRF2 modules are shown in *Figure 9* and *Figure 10* respectively. The tables in this chapter describe the control registers of the TRF1 and TRF2 modules.

Table 11: TRF(1, 2) configuration memory

Address (15 bits)	Bits	Description

0x0100	15	EN_LOWBWLOMX_TMX_TRF_(1, 2): Controls the high pass pole frequency of the RC biasing the gate of the mixer switches.
		      0 – High band – bias resistor 3K **(default)**
		      1 – Low band – bias resistor 30K

	14	EN_NEXTTX_TRF_(1, 2): Enables the daisy change LO buffer going from TRF_1 to TRF2
		      0 – Buffer disabled (SISO) **(default)**
		      1 – Buffer enabled (MIMO)

	13 – 12	EN_AMPHF_PDET_TRF_(1, 2)[1:0]: Enables the TXPAD power detector preamplifier
		      3 – Preamp gain 25dB **(default)**
		      2 – Do not use
		      1 – Preamp gain 7dB
		      0 – Preamp gain -10dB

	11 – 10	LOADR_PDET_TRF_(1, 2) [1:0]: Controls the resistive load of the Power detector
		      0 – R_DIFF 5K||2.5K||1.25K
		      1 – R_DIFF 5K||1.25K **(default)**
		      2 – R_DIFF 5K||2.5K
		      3 – R_DIGG 5K

	9 – 4	**Reserved**

	3	PD_PDET_TRF_(1, 2): Power down signal for Power Detector
		      0 – Enabled
		      1 – Powered down **(default)**

	2	PD_TLOBUF_TRF_(1, 2): Power down signal for TX LO buffer
		      0 – Enabled **(default)**
		      1 – Powered down

	1	PD_TXPAD_TRF_(1, 2): Power down signal for TXPAD
		      0 – Enabled **(default)**
		      1 – Powered down

	0	EN_G_TRF_(1, 2): Enable control for all the TRF_1 power downs
		      0 – All TRF_1 modules powered down
		      1 – All TRF_1 modules controlled by individual power down registers **(default)**
		
		Default: 00110100 00001001

0x0101	15 – 13	F_TXPAD_TRF_(1, 2)[2:0]: controls the switched capacitor at the TXPAD output. Is used for fine tuning of the TXPAD output. **Default: 3**

	12 – 11	L_LOOPB_TXPAD_TRF_(1, 2)[1:0]: Controls the loss of the of the loopback path at the TX side
		      0 – Loss=0dB
		      1 – Loss=20*log10(5)
		      2 – Loss=20*log10(11)
		      3 – Loss=20*log10(16) **(default)**

	10 – 6	LOSS_LIN_TXPAD_TRF_(1, 2)[4:0]: Controls the gain of the linearizing part of the TXPAD **Default: 0**
		      0<=Loss<=10 – Pout=Pout_max-Loss
		      11<=Loss<31 – Pout=Pout_max-10-2*(Loss-10)
		      Ideally LOSS_LIN = LOSS_MAIN

	5 – 1	LOSS_MAIN_TXPAD_TRF_(1, 2)[4:0]: Controls the gain & output power of the TXPAD. **Default: 0**
		      0<=Loss<=10 – Pout=Pout_max-Loss
		      11<=Loss<31 – Pout=Pout_max-10-2*(Loss-10)

	0	EN_LOOPB_TXPAD_TRF_(1, 2): Enables the TXPAD loopback path
		      0 – Loopback disabled **(default)**
		      1 – Loopback enabled
		
		Default: 01111000 00000000

Address (15 bits)	Bits	Description
0x0102	15	GCAS_GNDREF_TXPAD_TRF_(1, 2): Controls if the TXPAD cascode transistor gate bias is referred to VDD or GND. 　　　0 – VDD referred **(default)** 　　　1 – GNDS referred
	14 – 10	ICT_LIN_TXPAD_TRF_(1, 2)[4:0]: Control the bias current of the linearization section of the TXPAD. **Default: 12** 　　　I_bias=I_bias_nominal * ICT/12
	9 – 5	ICT_MAIN_TXPAD_TRF_(1, 2)[4:0]: Control the bias current of the main gm section of the TXPAD. **Default: 12** 　　　I_bias=I_bias_nominal * ICT/12
	4 – 0	VGCAS_TXPAD_TRF_(1, 2)[4:0]: Controls the bias voltage at the gate of TXPAD cascade. **Default: 0** 　　　vgcas=(VGCAS_TXOAD/12)*100u*10K, when GCAS_GNDREF=1 　　　vgcas=VDD18-(VGCAS_TXOAD/12)*100u*7.5K,　　　　　　when GCAS_GNDREF=0 **Default**: 00110001 10000000
0x0103	15 –12	**Reserved**
	11	SEL_BAND1_TRF_(1, 2): Enable signal for TXFE, band 1 　　　0 – Disabled 　　　1 – Enabled **(default)**
	10	SEL_BAND2_TRF_(1, 2): Enable signal for TXFE, band 2 　　　0 – Disabled **(default)** 　　　1 – Enabled
	9 – 5	LOBIASN_TXM_TRF_(1, 2)[4:0]: Controls the bias at the gate of the mixer NMOS switch. **Default: 16** 　　　Vgate_bias=Vth_nmos+25K*LOBIASN/12*20u
	4 – 0	LOBIASP_TXX_TRF_(1, 2)[4:0]: Controls the bias at the gate of the mixer PMOS switch. **Default: 18** 　　　Vgate_bias=Vth_pmos-25K*LOBIASP/12*20u **Default**: 00001010 00010010
0x0104	15 – 8	**Reserved**
	7 – 4	CDC_I_TRF_(1,2)[3:0]: Trims the duty cycle in I channel. **Default = 8;**
	3 – 0	CDC_Q_TRF_(1,2)[3:0]: Trims the duty cycle in Q channel. **Default = 8;** **Default**: 00000000 10001000

2.11 TBB(1, 2) Configuration Memory

The block diagrams of the TBB1 and TBB2 modules are shown in *Figure 11* and *Figure 12* respectively. The tables in this chapter describe the control registers of the TBB1 and TBB2 modules.

Table 12: TBB(1, 2) configuration memory

Address (15 bits)	Bits	Description
0x0105	15	STATPULSE_TBB_(1, 2): This is a narrow start-up pulse of more than 1us width. **Default: 0**
	14 – 12	LOOPB_TBB_(1, 2)[2:0]: This controls which signal is connected to the loopback output pins loopb as follows: Bits [1:0]: 0 – output is disconnected (high impedance) loop back is disabled. **(default)** 1 – DAC current output is routed to the loopb pins. 2 – low band ladder output is routed to the output. 3 – main TBB output is routed to the loopb outputs. Bit [2] (swaps the I Q channels): 0 TBB output I goes to loopb_2 path and Q goes to loopb_1 path. **(default)** 1 – TBB output I goes to loopb_1 path and Q goes to loopb_2 path. Note: when both the lowpass ladder and real pole are powered down, the output of the active highband biquad is routed to the loopb outputs on setting 3.
	11 – 5	**Reserved**
	4	PD_LPFH_TBB_(1, 2): This selectively powers down the LPFH_TBB biquad. Please note, the LPFH_TBB is powered down if any of the following is true: PD_LPFLAB_TBB=0 & PD_LPFS5_TBB=0, or, PD_TBB = 1, or PD_LPFH_TBB = 1. 0 – Active **(default)** 1 – powered down
	3	PD_LPFIAMP_TBB_(1, 2): This selectively powers down the LPFIAMP_TBB front-end current amp of the transmitter base band. Please note, the LPFIAMP_TBB is powered down if any of the following is true: PD_TBB = 1, or PD_LPFIAMP_TBB = 1 0 – Active **(default)** 1 – powered down
	2	PD_LPFLAD_TBB_(1, 2): This selectively powers down the LPFLAD_TBB low pass ladder filter of the transmitter base band. Please note, the ladder is powered down if any of the following is true: PD_TBB = 1, or PD_LPFLAD_TBB = 1 0 – Active 1 – powered down **(default)**
	1	PD_LPFS5_TBB_(1, 2): This selectively powers down the LPFS5_TBB low pass real-pole filter of the transmitter base band. Please note, the real-pole stage is powered down if any of the following is true: PD_TBB = 1, or PD_LPFS5_TBB = 1 0 – Active 1 – powered down **(default)**
	0	EN_G_TBB_(1, 2): Enable control for all the TBB_TOP power downs 0 – All TBB_TOP modules powered down 1 – All TBB_TOP modules may be selectively turned off **(default)** **Default**: 00000000 00000111

Address (15 bits)	Bits	Description
0x0106	15	**Reserved**
	14 – 10	ICT_LPFS5_F_TBB_(1, 2)[4:0]: This controls the operational amplifier's output stage bias current of the low band real pole filter of the transmitter's base band. **Default: 12**
	9 – 5	ICT_LPFS5_PT_TBB_(1, 2)[4:0]: This controls the operational amplifier's input stage bias current of the low band real pole filter of the transmitter's base band. **Default: 12**
	4 – 0	ICT_LPF_H_PT_TBB_(1, 2)[4:0]: This controls the operational amplifiers input stage bias reference current of the high band low pass filter of the transmitter's base band (LPFH_TBB). **Default: 12** **Default: 00110001 10001100**
0x0107	15	**Reserved**
	14 – 10	ICT_LPFH_F_TBB_(1, 2)[4:0]: This controls the operational amplifiers output stage bias reference current of the high band low pass filter of the transmitter's base band (LPFH_TBB). **Default: 12**
	9 – 5	ICT_LPFLAD_F_TBB_(1, 2)[4:0]: This controls the operational amplifiers' output stages bias reference current of the low band ladder filter of the transmitter's base band. **Default: 12**
	4 – 0	ICT_LPFLAD_PT_TBB_(1, 2)[4:0]: This controls the operational amplifiers' input stages bias reference current of the low band ladder filter of the transmitter's base band. **Default: 12** **Default: 00110001 10001100**

0x0108	15 –10	CG_IAMP_TBB_(1, 2)[5:0]: This controls the front-end gain of the TBB. For a given gain value, this control value varies with the set TX mode. After resistance calibration, the following table gives the nominal values for each frequency setting. However, this table is to be updated and corrected after calibration. **Default: 37**
		 Low Band:
		     5 – when 2.4 MHz
		     7 – when 2.74 MHz
		     12 – when 5.5 MHz
		     18 – when 8.2 MHz
		     24 – when 11 MHz
		 High Band:
		     18 – when 18.5 MHz
		     37 – when 38 MHz
		     54 – when 54 MHz

	9 – 5	ICT_IAMP_FRP_TBB_(1, 2)[4:0]: This controls the reference bias current of the IAMP main bias current sources. **Default: 12**
	4 – 0	ICT_IAMP_GG_FRP_TBB_(1, 2)[4:0]: This controls the reference bias current of the IAMP's cascode transistors gate voltages that set the IAMP's input voltage level. The IAMP's input is connected to the DAC output. **Default: 12** **Default: 10010101 10001100**

0x0109	15 – 8	RCAL_LPFH_TBB_(1, 2)[7:0]: This controls the value of the equivalent resistance of the resistor banks of the biquad filter stage (of the high band section) of the transmitter base band(TBB). **Default: 97**
		 Following is a nominal values table that are corrected for any process variation after calibration:
		     18 – when 18.5 MHz
		     97 – when 38 MHz
		     164 – when 54 MHz

	7 – 0	RCAL_LPFLAD_TBB_(1, 2)[7:0]: This controls the value of the equivalent resistance of the resistor banks of the low pass filter ladder (of the low band section) of the transmitter base band (TBB). **Default: 193**
		 Following is a nominal values table that are corrected for any process variations after calibration.
		     6 – when 2.4 MHz
		     19 – when 2.74 MHz
		     75 – when 5.5 MHz
		     133 – when 8.2 MHz
		     193 – when 11 MHz
		
		 Default: 01100001 11000001

Address (15 bits)	Bits	Description

0x010A	15 – 14	TSTIN_TBB_(1, 2)[1:0]: This control selects where the input test signal (vinp/n_aux_bbq/i) is routed to as well as disabling the route.
		    0 – Disabled. Test signal is not routed anywhere. (**default**)
		    1 – Test signal is routed to the input of the Highband Filter.
		    2 – Test signal is routed to the input of the LowBand Filter.
		    3 - Test signal is routed to the input of the current amplifier.

	13	BYPLADDER_TBB_(1, 2): This signal by passes the LPF ladder of TBB and directly connects the output of the current amplifier to the null port of the real pole stage of the TBB low pass filter.
		    1 – bypass is active
		    0 – bypass is inactive **(default)**

	12 – 8	CCAL_LPFLAD_TBB_(1, 2)[4:0]: A common control signal for all the capacitor banks of TBB filters (including the ladder, real pole, and the high band biquad). It is the calibrated value of the banks control that sets the value of the banks' equivalent capacitor to their respective nominal values. **Default: 16**
	7 – 0	RCAL_LPFS5_TBB_(1, 2)[7:0]: This controls the value of the equivalent resistance of the resistor banks of the real pole filter stage (of the low band section) of the transmitter base band (TBB). The following is a nominal values table that are corrected for any process variation after calibration: If >5.5 MHz 200 otherwise 76. **Default: 76** **Default**: 00010000 01001100
0x010B	15 – 6	**Reserved**
	5 – 0	RESRV_TBB_(1, 2)[5:0]: Reserved for future use. **Default: 0** **Default**: 00000000 00000000

2.12 AFE Configuration Memory

The block diagram of the AFE module is shown in *Figure 13*. The tables in this chapter describe the control registers of the AFE module.

Table 13: AFE configuration memory

Address (15 bits)	Bits	Description
0x0082	15 – 13	ISEL_DAC_AFE[2:0]: Controls the peak current of the DAC output current. **Default: 4** Iout_peak = 325 uA+ISEL_DAC_AFE*75 uA Nominal = 625 uA
	12	MODE_INTERLEAVE_AFE: time interleaves the two ADCs into one ADC 0 – Two ADCs **(default)** 1 – Interleaved
	11 – 10	MUX_AFE_1<1:0>: Controls the MUX at the input of the ADC channel 1 0 – MUX off, only PGA output is connected to ADC input **(default)** 1 – pdet_1 is connected to ADC channel 1. PGA should be powered down 2 – BIAS_TOP test outputs will be connected to ADC channel 1 input (Please see MUX_BIAS_OUT<1:0>) 3 – RSSI 1 output will be connected to ADC 1 input
	9 – 8	MUX_AFE_2<1:0>: Controls the MUX at the input of the ADC channel 2 0 – MUX off, only PGA output is connected to ADC input **(default)** 1 – pdet_2 is connected to ADC channel 2. PGA should be powered down 2 – RSSI 1 output will be connected to ADC 2 input 3 – RSSI 2 output will be connected to ADC 2 input
	7 – 6	**Reserved**
	5	PD_AFE: Power down control for the AFE current mirror in BIAS_TOP 0 – Active **(default)** 1 – powered down
	4	PD_RX_AFE1: Power down control for the ADC of channel 1 0 – Active **(default)** 1 – powered down
	3	PD_RX_AFE2: Power down control for the ADC of channel 2 0 – Active 1 – powered down **(default)**
	2	PD_TX_AFE1: Power down control for the DAC of channel 1 0 – Active **(default)** 1 – powered down
	1	PD_TX_AFE2: Power down control for the DAC of channel 2 0 – Active 1 – powered down **(default)**
	0	EN_G_AFE: Enable control for all the AFE power downs 0 – All AFE modules are powered down 1 – All AFE modules are controlled by individual power down registers **(default)** **Default**: 10000000 00001011

2.13 BIAS Configuration Memory

The block diagram of the BIAS module is shown in *Figure 14*. The tables in this chapter describe the control registers of the BIAS module.

Table 14: BIAS configuration memory

Address (15 bits)	Bits	Description
0x0083	15 – 11	**Reserved**
	10 – 0	RESRV_BIAS[10:0]: Reserve. **Default: 0**
		Default: 00000000 00000000
0x0084	15 – 13	**Reserved**

	12 – 11	MUX_BIAS_OUT[1:0]: Test mode of the BIAS_TOP
		        0 – NO test mode **(default)**
		        1 – vr_ext_bak and vr_cal_ref=600mV is passed to the ADC input MUX. Vr_ext_bak is the voltage read on the off-chip 10 kohm reference resistor. Ip60f is connected to r_ext=10 kohm and RP_CALIB_BIAS is changed until vr_ext becomes 600 mV.
		        2 – Vptat_600 mV and vr_cal_ref=600 mV is passed to the ADC input MUX. The ratio between the two will be proportional to the absolute temp.
		        3 – No test mode

	10 – 6	RP_CALIB_BIAS[4:0]: Calibration code for rppolywo. This code is set by the calibration algorithm: BIAS_RPPOLY_calibration **Default: 16**
	5	**Reserved**

	4	PD_FRP_BIAS: Power down signal for Fix/RP block
		        0 – Enabled **(default)**
		        1 – Powered down

	3	PD_F_BIAS: Power down signal for Fix
		        0 – Enabled **(default)**
		        1 – Powered down

	2	PD_PTRP_BIAS: Power down signal for PTAT/RP block
		        0 – Enabled **(default)**
		        1– Powered down

	1	PD_PT_BIAS: Power down signal for PTAT block
		        0 – Enabled **(default)**
		        1 – Powered down

	0	PD_BIAS_MASTER: Enable signal for central bias block
		        0 – Sub blocks may be selectively powered down **(default)**
		        1 – Powers down all BIAS blocks

		Default: 00000100 00000000

2.14 SXR, SXT Configuration Memory

The block diagrams of the SXT and SXR modules are shown in Figure 15 and Figure 16 respectively. The tables in this chapter describe the control registers of the SXR and SXT modules.

Table 15: SXT (SXR) configuration memory

Address (15 bits)	Bits	Description

0x011C	15	RESET_N_(SXR, SXT): Resets SX. A pulse should be used in the start-up to reset
		    0 – Reset
		    1 – Normal operation **(default)**

	14	SPDUP_VCO_(SXR, SXT): Bypasses the noise filter resistor for fast settling time. It should be connected to a 1 us pulse
		    0 – speed up disabled (noise filter resistor active) **(default)**
		    1 – speed up enabled (noise filter resistor shorted)

	13	BYPLDO_VCO_(SXR, SXT): Controls the bypass signal for the SX LDO
		    0 – LDO active
		    1 – LDO bypassed (input/output of the SX LDO shorted) **(default)**

	12	EN_COARSEPLL_(SXR, SXT): Enable signal for coarse tuning block
		    0 – Coarse tuning disabled **(default)**
		    1 – Coarse tuning enabled

	11	CURLIM_VCO_(SXR, SXT): Enables the output current limitation in the VCO regulator
		    0 – Current limit disabled
		    1 – Current limit enabled **(default)**

	10	EN_DIV2_DIVPROG_(SXR, SXT): Enables additional DIV2 prescaler at the input of the Programmable divider. The core of programmable divider in the SX feedback divider works up to 5.5 GHz. For FVCO>5.5 GHz, the prescaler is needed to lower the input frequency to DIVPROG_SX. **Shadow register.**
		    0 – DIVPROG input =
		    Fvco [Fvco=Fref*((INT_SDM_SX+4)+FRAC_SDM)
		    1 – DIVPROG input =
		    Fvco/2 [Fvco=2*Fref*((INT_SDM_SX+4)+FRAC_SDM) **(default)**

	9	EN_INTONLY_SDM_(SXR, SXT): Enables INTEGER-N mode of the SX
		    0 – Frac-N mode **(default)**
		    1 – INT-N mode

	8	EN_SDM_CLK_(SXR, SXT): Enables/Disables SDM clock. In INT-N mode or for noise testing, SDM clock can be disabled
		    0 – SDM clock disabled
		    1 – SDM clock enabled **(default)**

	7	PD_FBDIV_(SXR, SXT): Power down the feedback divider block.
		    0 – block active **(default)**
		    1 – block powered down

	6	PD_LOCH_T2RBUF: Power down for the LO buffer from SXT to SXR. To be active only in the TDD mode. **For Tx Path only!**
		    0 – block active
		    1 – block powered down **(default)**

	5	PD_CP_(SXR, SXT): Power down for the Charge Pump
		    0 – block active **(default)**
		    1 – block powered down

	4	PD_FDIV_(SXR, SXT): Power down for the forward frequency divider and divider chain of the LO chain.
		    0 – blocks active **(default)**
		    1 – blocks powered down

	3	PD_SDM_(SXR, SXT): Power down for the SDM
		    0 – block active **(default)**
		    1 – block powered down

	2	PD_VCO_COMP_(SXR, SXT): Power down for the VCO comparator
		    0 – block active **(default)**
		    1 – block powered down

	1	PD_VCO_(SXR, SXT): Power down for the VCO
		    0 – block active
		    1 – block powered down **(default)**

	0	EN_G_(SXR, SXT): Enable control for all the SX power downs
		    0 – All SXT modules powered down
		    1 – All SXT modules controlled by individual power down registers **(default)**
		
		Default: 10101101 01000011

Address (15 bits)	Bits	Description
0x011D	15 – 0	FRAC_SDM_(SXR, SXT)[15:0]: Fractional control of the division ratio LSB. **Default: 1024** =2^20*[Fvco/(Fref * 2^ EN_DIV2_DIVPROG_(SXR, SXT)) – int(Fvco/(Fref * 2^ EN_DIV2_DIVPROG_(SXR, SXT)))] **Default**: 00000100 00000000
0x011E	15 - 14	**Reserved**
	13 – 4	INT_SDM_(SXR, SXT)[9:0]: Controls Integer section of the division ratio INT_SDM_(SXR, SXT) = int(Fvco/2^(EN_DIV2_DIVPROG_(SXR, SXT))/Fref)-4 **Default: 120**
	3 – 0	FRAC_SDM_(SXR, SXT)[19:16]: Fractional control of the division ratio MSB. **Default**: 00000111 10000000
0x011F	15	**Reserved**
	14 – 12	PW_DIV2_LOCH_(SXR, SXT)[2:0]: trims the duty cycle of DIV2 LOCH. Only works when the forward divider is dividing by at least 2 (excluding quadrature block division). If in bypass mode, this does not work. **Default: 3**
	11 – 9	PW_DIV4_LOCH_(SXR, SXT)[2:0]: trims the duty cycle of DIV4 LOCH. Only works when the forward divider is dividing by at least 4 (excluding quadrature block division). If in bypass mode, this does not work. **Default: 3**
	8 – 6	DIV_LOCH_(SXR, SXT)[2:0]: Controls the division ratio in the LOCH_DIV. There is an additional DIV/2 in the quadrature generator → Flo = Fvco / divRatio_LOCH / 2 divRatio_LOCH = 2^(DIV_LOCH_SX) Note: Value 111 not allowed. **Shadow register. Default: 1**
	5 – 3	TST_SX_(SXR, SXT)[2:0]: Controls the test mode of PLLs. TST signal lines are shared between all PLLs (CGEN, RX and TX). Only one TST signal of any PLL should be active at a given time. 0 – TST disabled; RSSI analog outputs enabled if RSSI blocks active and when all PLL test signals are off **(default)** 1 – tstdo<0>=CLKH1 & tstdo<1>=CLKH2 2 – tstdo<0>=CLK_SDM & tstdo<1>=DIV_CLK 3 – tstdo<0>=REFCLK & tstdo<1>=DIV_CLK 2 – vco_tune-->tstao<0> through a 50Kohm resistor 3 – vco_tune-->tstao<0> through a 10Kohm resistor 5 – PFD UP --> tstdo<0> and PFD DN --> tstdo<1>
	2	if TST_SX<2>=1 --> VCO_TSTBUF active generating VCO_TST_DIV20 and VCO_TST_DIV40 SEL_SDMCLK_(SXR, SXT): Selects between the feedback divider output and Fref for SDM
	1	0 – CLK CLK_DIV **(default)** 1 – CLK CLK_REF SX_DITHER_EN_(SXR, SXT): Enabled dithering in the SDM
	0	0 – Disabled **(default)** 1 – Enabled REV_SDMCLK_(SXR, SXT): Reverses the SDM clock 0 – **(default)** 1 – reversed (after INV) **Default**: 00110110 01000000
0x0120	15 – 8	VDIV_VCO_(SXR, SXT)[7:0]: Controls the VCO LDO output voltage. **Default: 185** Vout(VCO_LDO)=VDD18_VCO* [29.1/(29.1 + 233/(VDIV_VCO_SX+2)))] 185 --> Vout(VCO_LDO)=1.55V (VDD18_VCO=1.72)
	7 – 0	ICT_VCO_(SXR, SXT)[7:0]: Scales the VCO bias current from 0 to 2.5xInom **Default: 128** **Default**: 10111001 10000000

Address (15 bits)	Bits	Description
0x0121	15 – 11	RSEL_LDO_VCO_(SXR, SXT)[4:0]: Set the reference voltage that supplies the bias voltage of the switch-cap array and varactor. **Default: 16** Vref=60 uA * 180 kohm / RSEL_LDO_VCO
	10 – 3	CSW_VCO_(SXR, SXT)[7:0]: coarse control of VCO frequency, 0 for lowest frequency and 255 for highest. This control is set by SX_SWC_calibration. **Shadow register. Default: 128**
	2 – 1	SEL_VCO_(SXR, SXT)[1:0]: Selects the active VCO. It is set by SX_SWC_calibration. **Shadow register.** 0 – VCOL 1 – VCOM 2 – VCOH **(default)** 3 – Not Valid
	0	COARSE_START_(SXR, SXT): Control signal for coarse tuning algorithm (SX_SWC_calibration). **Default: 0** **Default**: 10000100 00000100
0x0122	15 – 13	**Reserved**
	12	REVPH_PFD_(SXR, SXT): Reverse the pulses of the PFD. It can be used to reverse the polarity of the PLL loop (positive feedback to negative feedback). **Default: 0**
	11 – 6	IOFFSET_CP_(SXR, SXT)[5:0]: Scales the offset current of the charge pump, 0-->63. This current is used in Fran-N mode to create an offset in the CP response and avoid the non-linear section. **Default: 20** ioffset=0.243 uA * IOFFSET_CP_SX ioffset/ipulse=4/(INT_SDM_SX+4) [First estimation]
	5 – 0	IPULSE_CP_(SXR, SXT)[5:0]: Scales the pulse current of the charge pump, 0-->63. **Default: 20** ipulse=2.312 uA * IPULSE_CP_SX **Default**: 00000101 00010100
0x0123	15	COARSE_STEPDONE_(SXR, SXT): **Read only.**
	14	COARSEPLL_COMPO_(SXR, SXT): **Read only.**
	13	VCO_CMPHO_(SXR, SXT): Compares the Vtune value to a predefined value of 920 mV. **Read only register**. 0 – Vtune voltage level is higher than CMPHO threshold voltage of 920mV 1 – Vtune voltage level is lower than CMPHO threshold voltage of 920mV
	12	VCO_CMPLO_(SXR, SXT): Compares Vtune value to a predefined value of 180mV. **Read only register**. 0 – Vtune voltage level is higher than CMPLO threshold voltage of 180mV 1 – Vtune voltage level is lower than CMPLO threshold voltage of 180mV
	11 – 8	CP2_PLL_(SXR, SXT)[3:0]: Controls the value of CP2 (cap from CP output to GND) in the PLL filter. **Default: 6** cp2=CP2_PLL_SX*6*387fF
	7 – 4	CP3_PLL_(SXR, SXT)[3:0]: Controls the value of CP3 (cap from VCO Vtune input to GND) in the PLL filter. **Default: 7** cp3=CP3_PLL_SX*6*980fF
	3 – 0	CZ_(SXR, SXT)[3:0]: Controls the value of CZ (Zero capacitor) in the PLL filter. **Default: 11** cz=CZ_PLL_SX*8*5.88pF **Default**: 00000110 01111011

Address (15 bits)	Bits	Description
0x0124	15 – 11	RESRV_(SXR, SXT)[4:0]: Reserved. **Default: 0**
		(For SXT only RESRV_SXR[0] **connected to the output!)**
	10 – 5	**Reserved**
	4	EN_DIR_(SXR, SXT): Enables direct control of PDs and ENs for SXR/SXT module.
		0 – direct control disabled **(default)**
		1 – direct control enabled
	3	EN_DIR_RBB(1, 2): Enables direct control of PDs and ENs for RBB(1, 2) module.
		0 – direct control disabled **(default)**
		1 – direct control enabled
	2	EN_DIR_RFE(1, 2): Enables direct control of PDs and ENs for RFE(1, 2) module.
		0 – direct control disabled **(default)**
		1 – direct control enabled
	1	EN_DIR_TBB(1, 2): Enables direct control of PDs and ENs for TBB(1, 2) module.
		0 – direct control disabled **(default)**
		1 – direct control enabled
	0	EN_DIR_TRF(1, 2): Enables direct control of PDs and ENs for TRF(1, 2) module.
		0 – direct control disabled **(default)**
		1 – direct control enabled
		Default: 00000000 00000000

2.15 CGEN Configuration Memory

The block diagram of the CGEN module is shown in *Figure 17*. The tables in this chapter describe the control registers of the CGEN module.

Table 16: CGEN configuration memory

Address (15 bits)	Bits	Description

0x0086	15	SPDUP_VCO_CGEN: Bypasses the noise filter resistor for fast settling time. It should be connected to a 1us pulse.
		        0 – speed up disabled (noise filter resistor active) **(default)**
		        1 – speed up enabled (noise filter resistor shorted)

	14	RESET_N_CGEN: Resets SX. A pulse should be used in the start-up to reset.
		        0 – Reset
		        1 – Normal operation **(default)**

	13 – 12	**Reserved**

	11	EN_ADCCLKH_CLKGN: Selects if F_CLKH or F_CLKL is connected to FCLK_ADC (F_CLKH and F_CLKL are the two internally generated clocks. A MUX will connect one of them to FCLK_ADC and the other to FCLK_DAC.).
		        0 – FCLK_ADC from F_CLKH / FCLK_DAC from F_CLKL
		        1 – FCLK_ADC from F_CLKL / FCLK_DAC from F_CLKH **(default)**

	10	EN_COARSE_CKLGEN: Enable signal for coarse tuning block.
		        0 – Coarse tuning disabled **(default)**
		        1 – Coarse tuning enabled

	9	EN_INTONLY_SDM_CGEN: Enables INTEGER-N mode of the SX.
		        0 – Frac-N mode **(default)**
		        1 – INT-N mode

	8	EN_SDM_CLK_CGEN: Enables/Disables the SDM clock. In INT-N mode or for noise testing, SDM clock can be disabled.
		        0 – SDM clock disabled
		        1 – SDM clock enabled **(default)**

	7	**Reserved**

	6	PD_CP_CGEN: Power down for the Charge Pump.
		        0 – block active **(default)**
		        1 – block powered down

	5	PD_FDIV_FB_CGEN: Power down for the feedback frequency divider.
		        0 – block active **(default)**
		        1 – block powered down

	4	PD_FDIV_O_CGEN: Power down for the forward frequency divider of the CGEN block.
		        0 – block active **(default)**
		        1 – block powered down

	3	PD_SDM_CGEN: Power down for the SDM.
		        0 – block active **(default)**
		        1 – block powered down

	2	PD_VCO_CGEN: Power down for the VCO.
		        0 – block active
		        1 – block powered down **(default)**

	1	PD_VCO_COMP_CGEN: Power down for the VCO comparator.
		        0 – block active **(default)**
		        1 – block powered down

	0	EN_G_CGEN: Enable control for all the CGEN power downs.
		        0 – All CGEN modules powered down
		        1 – All CGEN modules controlled by individual power down registers **(default)**
		
		Default: 01001001 00000101

0x0087	15 – 0	FRAC_SDM_CGEN[15:0]: Fractional control of the division ratio LSB. **Default: 1024** =2^20*[Fvco/Fref - int(Fvco/Fref)] **Default**: 00000100 00000000
0x0088	15 - 14	**Reserved**
	13 – 4	INT_SDM_CGEN [9:0]: Controls Integer section of the division ratio **Default: 120** INT_SDM_SX=int(Fvco/Fref)-1
	3 – 0	FRAC_SDM_CGEN [19:16]: Fractional control of the division ratio MSB. **Default**: 00000111 10000000

Address (15 bits)	Bits	Description

0x0089	15	REV_SDMCLK_CGEN: Reverses the SDM clock
		    0 – default **(default)**
		    1 – reversed (after INV)

	14	SEL_SDMCLK_CGEN: Selects between the feedback divider output and Fref for SDM
		    0 – CLK CLK_DIV **(default)**
		    1 – CLK CLK_REF

	13	SX_DITHER_EN_CGEN: Enabled dithering in the SDM
		    0 – Disabled **(default)**
		    1 – Enabled

	12 – 11	CLKH_OV_CLKL_CGEN[1:0]: FCLKL here is the ADC clock. FCLKH is the clock to the DAC and if no division is added to the ADC as well. **Default: 0** FCLKL=FCLKH/2^(CLKH_OV_CLKL)
	10 – 3	DIV_OUTCH_CGEN[7:0]: Controls the output divider chain of the CGEN. F_CLKH=Fvco_CGEN/(2*(DIV_OUTCH_CGEN+1)) **Shadow register.** **Default: 4**

	2 – 0	TST_CGEN[2:0]: Controls the test mode of the SX
		Controls the test mode of the SX
		    0: TST disabled; RSSI analog outputs enabled if RSSI blocks active and when all PLL test signals are off **(default)**
		    1: tstdo[0]=REFCLK & tstdo[1]=DIV_OUT & tstdo[2]=CLK_SDM
		    2: vco_tune-->tstao[0] through a 50 kohm resistor
		    3: vco_tune-->tstao[0]
		    5: PFD UP --> tstdo[0] and PFD DN --> tstdo[1]
		    if TST_SX[2]=1 --> VCO_TSTBUF active generating VCO_TST_DIV20 and VCO_TST_DIV40
		
		Default: 00000000 00100000

0x008A	15	**Reserved**

	14	REV_CLKDAC_CGEN: Inverts the clock F_CLKL.
		    0 – Normal **(default)**
		    1 – Inverted

	13	REV_CLKADC_CGEN: Inverts the clock F_CLKL.
		    0 – Normal **(default)**
		    1 – Inverted

	12	REVPH_PFD_CGEN: Reverse the pulses of the PFD. It can be used to reverse the polarity of the PLL loop (positive feedback to negative feedback). **Default: 0**

	11 – 6	IOFFSET_CP_CGEN[5:0]: Scales the offset current of the charge pump, 0-->63. This current is used in Fran-N mode to create an offset in the CP response and avoid the non-linear section. **Default: 20**
		    ioffset=0.243uA * IOFFSET_CP_SX
		    ioffset/ipulse=4/(INT_SDM_SX+4) [First estimation]

	5 – 0	IPULSE_CP_CGEN[5:0]: Scales the pulse current of the charge pump, 0-->63. **Default: 20** ipulse=2.312uA * IPULSE_CP_SX **Default**: 00000101 00010100
0x008B	15 – 14	**Reserved**
	13 – 9	ICT_VCO_CGEN[4:0]: Scales the VCO bias current from 0 to 2.5xInom. **Default: 12**
	8 – 1	CSW_VCO_CGEN[7:0]: coarse control of VCO frequency, 0 for lowest frequency and 255 for highest. This control is set by SX_SWC_calibration. **Shadow register. Default: 128**
	0	COARSE_START_CGEN: Control signal for coarse tuning algorithm (SX_SWC_calibration). **Default: 0** **Default**: 00011001 00000000

Address (15 bits)	Bits	Description
0x008C	15	COARSE_STEPDONE_CGEN: **Read only**
	14	COARSEPLL_COMPO_CGEN: **Read only**
	13	VCO_CMPHO_CGEN: **Read only**
	12	VCO_CMPLO_CGEN: **Read only**
	11 – 8	CP2_CGEN[3:0]: Controls the value of CP2 (cap from CP output to GND) in the PLL filter. **Default: 6** cp2=CP2_PLL_SX*6*63.2fF
	7 – 4	CP3_CGEN[3:0]: Controls the value of CP3 (cap from VCO Vtune input to GND) in the PLL filter. **Default: 7** cp3=CP3_PLL_SX*248fF
	3 – 0	CZ_CGEN[3:0]: Controls the value of CZ (Zero capacitor) in the PLL filter. **Default: 11** cz=CZ_PLL_SX*8*365fF **Default**: 00000110 01111011
0x008D	15 – 3 2 – 0	**Reserved** RESRV_CGN[3:1]: Reserved **Default: 0** **Default**: 00000000 00000000

2.16 XBUF Configuration Memory

The block diagram of the XBUF module is shown in *Figure 18*. The tables in this chapter describe the control registers of the XBUF module.

Table 17: XBUF configuration memory

Address (15 bits)	Bits	Description
0x0085	15 – 9	**Reserved**
	8	SLFB_XBUF_RX: Self biasing digital control. 1 – enable biasing the input's DC voltage level from the chip, the input signal, IN, needs to be AC coupled to the chip 0 – disable the DC voltage level from the chip, the input signal, IN, needs to be DC coupled to the chip **(default)**
	7	SLFB_XBUF_TX: Self biasing digital control. 1 – enable biasing the input's DC voltage level from the chip, the input signal, IN, needs to be AC coupled to the chip. 0 – disable the DC voltage level from the chip, the input signal, IN, needs to be DC coupled to the chip. **(default)**
	6	BYP_XBUF_RX: Shorts the Input 3.3V buffer in XBUF The final 2 1.2V buffers are still active. The input in Bypass mode should be a 1.2 V full scale CMOS signal. 0 – Bypass not active **(default)** 1 – Bypass active
	5	BYP_XBUF_TX: Shorts the Input 3.3V buffer in XBUF The final two 1.2 V buffers are still active. The input in Bypass mode should be a 1.2 V full scale CMOS signal. 0 – Bypass not active **(default)** 1 – Bypass active
	4	EN_OUT2_XBUF_TX: Enables the 2nd output of TX XBUF. This 2nd buffer goes to XBUF_RX. This should be active when only 1 XBUF is to be used. 0 – TX XBUF 2nd output is active **(default)** 1 – TX XBUF 2nd output is disabled
	3	EN_TBUFIN_XBUF_RX: Disables the input from the external XOSC and buffers the 2nd input signal (from TX XBUF 2nd output) to the RX. This should be active when only 1 XBUF is to be used. 0 – RX XBUF input is coming from external XOSC **(default)** 1 – RX XBUF input is coming from TX
	2	PD_XBUF_RX: Power down signal 0 – block active **(default)** 1 – block powered down
	1	PD_XBUF_TX: Power down signal 0 – block active **(default)** 1 – block powered down
	0	EN_G_XBUF: Enable control for all the XBUF power downs 0 – All XBUF modules powered down 1 – All XBUF modules controlled by individual power down registers **(default)** **Default**: 00000000 00000001

2.17 LDO Configuration Memory

The block diagram of the LDO module is shown in *Figure 19*. The tables in this chapter describe the control registers of the LDO modules.

Table 18: LDO configuration memory

Address (15 bits)	Bits	Description

0x0092	15	EN_LDO_DIG: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	14	EN_LDO_DIGGN: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	13	EN_LDO_DIGSXR: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	12	EN_LDO_DIGSXT: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	11	EN_LDO_DIVGN: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	10	EN_LDO_DIVSXR: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	9	EN_LDO_DIVSXT: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	8	EN_LDO_LNA12: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	7	EN_LDO_LNA14: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	6	EN_LDO_MXRFE: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	5	EN_LDO_RBB: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	4	EN_LDO_RXBUF: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	3	EN_LDO_TBB: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	2	EN_LDO_TIA12: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	1	EN_LDO_TIA14: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	0	EN_G_LDO: Enable control for all the LDO power downs
		    0 – All LDO modules powered down
		    1 – All LDO modules controlled by individual power down registers **(default)**
		
		Default: 00000000 00000001

Address (15 bits)	Bits	Description

0x0093	15	EN_LOADIMP_LDO_TLOB: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	14	EN_LOADIMP_LDO_TPAD: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	13	EN_LOADIMP_LDO_TXBUF: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	12	EN_LOADIMP_LDO_VCOGN: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	11	EN_LOADIMP_LDO_VCOSXR: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	10	EN_LOADIMP_LDO_VCOSXT: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	9	EN_LDO_AFE: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	8	EN_LDO_CPGN: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	7	EN_LDO_CPSXR: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	6	EN_LDO_TLOB: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	5	EN_LDO_TPAD: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	4	EN_LDO_TXBUF: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	3	EN_LDO_VCOGN: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	2	EN_LDO_VCOSXR: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	1	EN_LDO_VCOSXT: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled

	0	EN_LDO_CPSXT: Enables the LDO
		    0 – Powered down **(default)**
		    1 – Enabled
		
		Default: 00000000 00000000

Address (15 bits)	Bits	Description
0x0094	15	EN_LOADIMP_LDO_CPSXT: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	14	EN_LOADIMP_LDO_DIG: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	13	EN_LOADIMP_LDO_DIGGN: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	12	EN_LOADIMP_LDO_DIGSXR: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	11	EN_LOADIMP_LDO_DIGSXT: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	10	EN_LOADIMP_LDO_DIVGN: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	9	EN_LOADIMP_LDO_DIVSXR: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	8	EN_LOADIMP_LDO_DIVSXT: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	7	EN_LOADIMP_LDO_LNA12: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	6	EN_LOADIMP_LDO_LNA14: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	5	EN_LOADIMP_LDO_MXRFE: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	4	EN_LOADIMP_LDO_RBB: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	3	EN_LOADIMP_LDO_RXBUF: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	2	EN_LOADIMP_LDO_TBB: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	1	EN_LOADIMP_LDO_TIA12: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias
	0	EN_LOADIMP_LDO_TIA14: Enables the load dependent bias to optimize the load regulation 0 – Constant bias **(default)** 1 – Load dependent bias **Default**: 00000000 00000000

Address (15 bits)	Bits	Description

0x0095	15	BYP_LDO_TBB: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	14	BYP_LDO_TIA12: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	13	BYP_LDO_TIA14: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	12	BYP_LDO_TLOB: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	11	BYP_LDO_TPAD: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	10	BYP_LDO_TXBUF: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	9	BYP_LDO_VCOGN: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	8	BYP_LDO_VCOSXR: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	7	BYP_LDO_VCOSXT: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	6 – 3	**Reserved**

	2	EN_LOADIMP_LDO_AFE: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	1	EN_LOADIMP_LDO_CPGN: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias

	0	EN_LOADIMP_LDO_CPSXR: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load dependent bias
		
		Default: 00000000 00000000

Address (15 bits)	Bits	Description

0x0096	15	BYP_LDO_AFE: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	14	BYP_LDO_CPGN: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	13	BYP_LDO_CPSXR: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	12	BYP_LDO_CPSXT: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	11	BYP_LDO_DIG: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	10	BYP_LDO_DIGGN: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	9	BYP_LDO_DIGSXR: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	8	BYP_LDO_DIGSXT: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	7	BYP_LDO_DIVGN: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	6	BYP_LDO_DIVSXR: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	5	BYP_LDO_DIVSXT: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	4	BYP_LDO_LNA12: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	3	BYP_LDO_LNA14: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	2	BYP_LDO_MXRFE: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	1	BYP_LDO_RBB: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	0	BYP_LDO_RXBUF: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput
		
		Default: 00000000 00000000

Address (15 bits)	Bits	Description
0x0097	15	SPDUP_LDO_DIVSXR: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	14	SPDUP_LDO_DIVSXT: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	13	SPDUP_LDO_LNA12: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	12	SPDUP_LDO_LNA14: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	11	SPDUP_LDO_MXRFE: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	10	SPDUP_LDO_RBB: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	9	SPDUP_LDO_RXBUF: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	8	SPDUP_LDO_TBB: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	7	SPDUP_LDO_TIA12: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	6	SPDUP_LDO_TIA14: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	5	SPDUP_LDO_TLOB: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	4	SPDUP_LDO_TPAD: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	3	SPDUP_LDO_TXBUF: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	2	SPDUP_LDO_VCOGN: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	1	SPDUP_LDO_VCOSXR: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up
	0	SPDUP_LDO_VCOSXT: Short the noise filter resistor to speed up the settling time 0 – noise filter resistor in place **(default)** 1 – Noise filter resistor bypassed should be connected to a 1~5 us pulse at the power up **Default**: 00000000 00000000

Address (15 bits)	Bits	Description
0x0098	15 – 9	**Reserved**
	8	SPDUP_LDO_AFE: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	7	SPDUP_LDO_CPGN: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	6	SPDUP_LDO_CPSXR: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	5	SPDUP_LDO_CPSXT: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	4	SPDUP_LDO_DIG: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	3	SPDUP_LDO_DIGGN: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	2	SPDUP_LDO_DIGSXR: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	1	SPDUP_LDO_DIGSXT: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up
	0	SPDUP_LDO_DIVGN: Short the noise filter resistor to speed up the settling time 　　　0 – noise filter resistor in place **(default)** 　　　1 – Noise filter resistor bypassed 　　　should be connected to a 1~5 us pulse at the power up **Default**: 00000000 00000000
0x0099	15 – 8	RDIV_VCOSXR[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_VCOSXT[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x009A	15 – 8	RDIV_TXBUF[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_VCOGN[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 140** 　　　Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 10001100
0x009B	15 – 8	RDIV_TLOB[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_TPAD[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x009C	15 – 8	RDIV_TIA12[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** 　　　Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_TIA14[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 140** 　　　Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 10001100

Address (15 bits)	Bits	Description
0x009D	15 – 8	RDIV_RXBUF[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_TBB[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x009E	15 – 8	RDIV_MXRFE[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_RBB[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 140** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 10001100
0x009F	15 – 8	RDIV_LNA12[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_LNA14[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 140** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 10001100
0x00A0	15 – 8	RDIV_DIVSXR[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_DIVSXT[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x00A1	15 – 8	RDIV_DIGSXT[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_DIVGN[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x00A2	15 – 8	RDIV_DIGGN[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_DIGSXR[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x00A3	15 – 8	RDIV_CPSXT[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_DIG[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x00A4	15 – 8	RDIV_CPGN[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_CPSXR[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101
0x00A5	15 – 8	RDIV_SPIBUF[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_AFE[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101

Address (15 bits)	Bits	Description
0x00A6	15 – 13	**Reserved**

	12	SPDUP_LDO_SPIBUF: Short the noise filter resistor to speed up the settling time
		    0 – Noise filter resistor in place **(default)**
		    1 – Noise filter resistor bypassed
		    should be connected to a 1~5 us pulse at the power up

	11	SPDUP_LDO_DIGIp2: Short the noise filter resistor to speed up the settling time
		    0 – Noise filter resistor in place **(default)**
		    1 – Noise filter resistor bypassed
		    should be connected to a 1~5 us pulse at the power up

	10	SPDUP_LDO_DIGIp1: Short the noise filter resistor to speed up the settling time
		    0 – Noise filter resistor in place **(default)**
		    1 – Noise filter resistor bypassed
		    should be connected to a 1~5 us pulse at the power up

	9	BYP_LDO_SPIBUF: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	8	BYP_LDO_DIGIp2: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	7	BYP_LDO_DIGIp1: Bypass signal for the LDO
		    0 – Does not bypass. Normal LDO operation **(default)**
		    1 – Bypasses LDO. Connects Vinput to Voutput

	6	EN_LOADIMP_LDO_SPIBUF: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load depdent bias

	5	EN_LOADIMP_LDO_DIGIp2: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load depdent bias

	4	EN_LOADIMP_LDO_DIGIp1: Enables the load dependent bias to optimize the load regulation
		    0 – Constant bias **(default)**
		    1 – Load depdent bias

	3	PD_LDO_SPIBUF: Enables the LDO
		    0 – Block active **(default)**
		    1 – Power down

	2	PD_LDO_DIGIp2: Enables the LDO
		    0 – Block active **(default)**
		    1 – Power down

	1	PD_LDO_DIGIp1: Enables the LDO
		    0 – Block active **(default)**
		    1 – Power down

	0	EN_G_LDOP: Enable control for all the LDO power downs
		    0 – All LDO modules powered down
		    1 – All LDO modules controlled by individual power down registers **(default)**
		
		Default: 00000000 00000001

0x00A7	15 – 8	RDIV_DIGIp2[7:0]: Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV
	7 – 0	RDIV_DIGIp1[7:0]:Controls the output voltage of the LDO by setting the resistive voltage divider ratio. **Default: 101** Vout=860 mV+3.92 mV *RDIV **Default**: 01100101 01100101

2.18 EN_DIR Configuration Memory

The tables in this chapters describe the control registers of the EN_DIR module. Each EN_DIR bit enables the capability of direct control of PD (powerdown) and EN (enable) outputs.

Table 19: EN_DIR configuration memory

Address (15 bits)	Bits	Description
0x0081	15 – 4	**Reserved**

	3	EN_DIR_LDO: Enables direct control of PDs and ENs for the LDO module.
		    0 – direct control disabled **(default)**
		    1 – direct control enabled

	2	EN_DIR_CGEN: Enables direct control of PDs and ENs for the CGEN module.
		    0 – direct control disabled **(default)**
		    1 – direct control enabled

	1	EN_DIR_XBUF: Enables direct control of PDs and ENs for the XBUF module.
		    0 – direct control disabled **(default)**
		    1 – direct control enabled

	0	EN_DIR_AFE: Enables direct control of PDs and ENs for the AFE module.
		    0 – direct control disabled **(default)**
		    1 – direct control enabled

		Default: 00000000 00000000

For other modules (SX (R, T), RBB (1, 2), RFE (1, 2), TBB (1, 2), TRF (1, 2)) EN_DIR can be controlled from register 0x0124.

2.19 SXR, SXT and CGEN BIST Configuration Memory

The block diagram of the BIST module for SXR, SXT and CGEN is shown in *Figure 24*. The table in this chapter describes the control registers of the BIST module.

There is one test vector generator which supplies the test vectors for the CGEN, SXT and SXR modules.

The register BSTART at 0x00A8[0] is used to initiate the BIST procedure for the selected modules. Registers BENC, BENR and BENT indicates which modules are to be tested. As an example, if BENC=1, BENR=0 and BENT=0 when BIST start is initiated, then the test procedure will be performed on SXR only. If BENC=1, BENR=1 and BENT=1 when BIST start is initiated, then BIST will be performed for the CGEN, SXR and SXT.

When BSTATE indicates the end of the BIST procedure, BSIGT, BSIGR and BSIGC registers will contain BIST signatures.

Table 20: BIST configuration memory

Address (15 bits)	Bits	Description
0x00A8	15 – 9	BSIGT[6:0]: BIST signature, Transmitter, LSB. **Default: 0**

	8	BSTATE: BIST state indicator **(read only)**
		          0 – BIST is not running **(default)**
		          1 – BIST in progress

	7	**Reserved**

	6	EN_SDM_TSTO_SXT: Enables the SDM_TSTO<12:0> outputs which will buffer the SDM outputs (inputs to the frequency divider) for testing purposes.
		          0 – all outputs are grounded **(default)**
		          1 – SDM_TSTO active

	5	EN_SDM_TSTO_SXR: Enables the SDM_TSTO<12:0> outputs which will buffer the SDM outputs (inputs to the frequency divider) for testing purposes.
		          0 – all outputs are grounded **(default)**
		          1 – SDM_TSTO active

	4	EN_SDM_TSTO_CGEN: Enables the SDM_TSTO<12:0> outputs which will buffer the SDM outputs (inputs to the frequency divider) for testing purposes.
		          0 – all outputs are grounded **(default)**
		          1 – SDM_TSTO active

	3	BENC: enables CGEN BIST
		          0 – disabled **(default)**
		          1 – enabled

	2	BENR: enables receiver BIST
		          0 – disabled **(default)**
		          1 – enabled

	1	BENT: enables transmitter BIST
		          0 – disabled **(default)**
		          1 – enabled

	0	BSTART: Starts delta sigma built in self test. Keep it at 1 one at least three clock cycles.
		          0 – **(default)**
		          0-to-1 – positive edge activates BIST
		
		Default: 00000000 00000000

0x00A9	15 – 0	BSIGT[22:7]: BIST signature, Transmitter, MSB **(read only)** **Default**: 00000000 00000000
0x00AA	15 – 0	BSIGR[15:0]: BIST signature, Receiver, LSB **(read only)** **Default**: 00000000 00000000
0x00AB	15 – 7	BSIGC[8:0]: BIST signature, CGEN , LSB **(read only)**
	6 – 0	BSIGR[22:16]: BIST signature, Receiver, MSB **(read only)** **Default**: 00000000 00000000
0x00AC	15 – 14	**Reserved**
	13 – 0	BSIGC[22:9]: BIST signature, CGEN , MSB **(read only)** **Default**: 00000000 00000000

2.20 CDS Configuration Memory

The block diagram of the Clock Distribution System (CDS) module is shown in *Figure 20*. The tables in this chapter describe the control registers of the CDS module.

Table 21: CDS configuration memory

Address (15 bits)	Bits	Description
0x00AD	15 – 14	CDS_MCLK2[1:0]: MCLK2 clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	13 – 12	CDS_MCLK1[1:0]: MCLK1 clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	11 – 10	**Reserved**
	9	CDSN_TXBTSP: TX TSPB clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	8	CDSN_TXATSP: TX TSPA clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	7	CDSN_RXBTSP: RX TSPB clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	6	CDSN_RXATSP: RX TSPA clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	5	CDSN_TXBLML: TX LMLB clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	4	CDSN_TXALML: TX LMLA clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	3	CDSN_RXBLML: RX LMLB clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	2	CDSN_RXALML: RX LMLA clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	1	CDSN_MCLK2: MCLK2 clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**)
	0	CDSN_MCLK1: MCLK1 clock inversion control. 0 – Clock is inverted 1 – Clock is not inverted (**default**) **Default**: 00000011 11111111

Address (15 bits)	Bits	Description
0x00AE	15 – 14	CDS_TXBTSP[1:0]: TX TSP B clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	13 – 12	CDS_TXATSP[1:0] : TX TSP A clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	11 – 1 0	CDS_RXBTSP[1:0]: RX TSP B clock delay. 00 – delay by 200 ps (**default**) 01 – delay by 500 ps 10 – delay by 800 ps 11 – delay by 1100 ps
	9 – 8	CDS_RXATSP[1:0]: RX TSP A clock delay. 00 – delay by 200 ps (**default**) 01 – delay by 500 ps 10 – delay by 800 ps 11 – delay by 1100 ps
	7 – 6	CDS_TXBLML[1:0]: TX LML B clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	5 – 4	CDS_TXALML[1:0]: TX LML A clock delay. 00 – delay by 400 ps (**default**) 01 – delay by 500 ps 10 – delay by 600 ps 11 – delay by 700 ps
	3 – 2	CDS_RXBLML[1:0]: RX LML B clock delay. 00 – delay by 200 ps (**default**) 01 – delay by 500 ps 10 – delay by 800 ps 11 – delay by 1100 ps
	1 – 0	CDS_RXALML[1:0]: RX LML A clock delay. 00 – delay by 200 ps (**default**) 01 – delay by 500 ps 10 – delay by 800 ps 11 – delay by 1100 ps **Default**: 00000000 00000000

2.21 mSPI Configuration Memory

More information about the embedded microcontroller may be found in the microcontroller datasheet.

Table 22: mSPI configuration memory

Address (15 bits)	Bits	Description
0x0000 Controls port P0 inputs (mSPI_REG0)	15 – 8 7 – 0	**Reserved** P0[7:0]: The data at MCU port P0 input can be changed by writing data into this register
0x0001 Reads port P1 outputs (mSPI_REG1) (read only)	15 – 8 7 – 0	**Reserved** P1[7:0]: The content of MCU P1 port output can be obtained by reading this register

0x0002
 	
Controls MCU input pins (mSPI_REG2)
	15 – 8
	7
	
	6
	
	5 – 2
	
	1 – 0
		Reserved
		RXD: The MCU USART receive input pin
		
		DEBUG: enables hardware MCU debugging mode
		      0 – normal mode
		      1 – debug mode
		
		EXT_INT[5:2]: external interrupts
		
		MODE[1:0]: controls MCU program memory initialization modes:
		      0 – the MCU is in reset
		      1 – Programming both EEPROM and SRAM through mSPI
		      2 – Programming only SRAM only through mSPI
		      3 – Programming SRAM by reading the EEPROM

0x0003 Reads MCU status signals (mSPI_REG3) (read only)	15 – 8 7 6 5-4 3 2 1 0	**Reserved** TXD: The USART transmit output pin PROGRAMMED: Status output signal; when set, it indicates that the programming process is finished, and the MCU executes instructions **Reserved** READ_REQ: status signal; new 8-bit data (the register mSPI_REG5 content) is ready to be read through the mSPI WRITE_REQ: status signal; a new data byte is waiting in the mSPI_REG4 register to be transferred into the MCU FULL_WRITE_BUFF: indicates that INPUT 32-byte FIFO buffer is full, the MCU is not ready to receive data, and the base band processor has to wait EMPTY_WRITE_BUFF: tells that INPUT 32-byte FIFO is empty
0x0004 Writes one byte of data to MCU (mSPI_REG4)	15 – 8 7 – 0	**Reserved** DTM[7:0]: output (byte) is fed to Data_to_MCU(7:0) input bus
0x0005 Reads data byte from MCU (mSPI_REG5) (read only)	15 – 8 7 – 0	**Reserved** DFM[7:0]: data (byte) received from bus Data_from_MCU(7:0)

0x0006
 	
Controls SPI switch (mSPI_REG6)
	15 – 1
	0
		Reserved
		SPISW_CTRL: controls the SPI switch
		      0 – Transceiver is controlled by the Base Band **(default)**
		      1 – Transceiver is controlled by the MCU

Appendix 1

SPI Procedures

A1.1 SPI READ/WRITE Pseudo Code

```
//-----------------------------------------------------------------------------
// Write command, SPI module address, register address
// Read data
//-----------------------------------------------------------------------------
void SPI_Read(unsigned int COMMAND)
{
        unsigned int DATA;     //We will read data there

        //Write Command and Address (MSB First)
//First 1 bit (MSB)  = Command
//Next 15 (LSBs) bits = Register Address
        for(int i=15; i>=0; i--)
        {
                if(i'th bit in COMMAND is '1')
                {
                        Set Data Output line to '1';
                }
                else
                {
                        Set Data Output line to '0';
                };
                Apply Rising and Falling CLK signal edges to CLK line;
        };

        //Read Data (MSB First)
        //Note: At this point we have data MSB valid from the chip.
        for(int i=15; i>=0; i--)
        {
                if(there is '1' at the Data Input Line)
                {
                        Set i'th bit in DATA '1';
                }
                else
                {
                        Set i'th bit in DATA '0';
                };
                Apply Rising and Falling CLK signal edges to CLK line;
        };
};
```

```
//----------------------------------------------------------------------------
// Write data to the chip:
// First byte: Command, SPI module address, register address
// Second byte: Data
//----------------------------------------------------------------------------
void SPI_Write(unsigned int COMMAND, unsigned int DATA)
{
        //Write Command, Address
        for(int i=15; i>=0; i--)
        {
                if(i'th bit in COMMAND is '1')
                {
                        Set Data Output line to '1';
                }
                else
                {
                        Set Data Output line to '0';
                };
                Apply Rising and Falling CLK signal edges to CLK line;
        };

        //Write Data
        for(int i=15; i>=0; i--)
        {
                if(i'th bit in DATA is '1')
                {
                        Set Data Output line to '1';
                }
                else
                {
                        Set Data Output line to '0';
                };
                Apply Rising and Falling CLK signal edges to CLK line;
        };
};
```

Appendix 2

Control Block Diagrams

A2.1 RFE Control Diagrams

Figure 5 RFE1 control structure

**LMS7002M
RXFE_TOP 2**

CDC_I_RFE_2 (0x010C[15:12])
CDC_Q_RFE_2 (0x010C[11:8])
SEL_PATH_RFE_2 (0x010D[8:7])
PD_QGEN_RFE_2 (0x010C[3])

From: RLODIVO_OUT1 (RX RF 1)

Always powered down!

LO I/Q generator

Grounded

RX RF loopback Buffer

LO mixer buffer

SEL_PATH_RFE_2 (0x010D[8:7])
PD_MXLOBUF_RFE_2 (0x010C[4])

Buffers for SPI signals

Bias blocks

From: rfg(p/n)_h_RFE_2 (PAD)
From: rfs(p/n)_h_RFE_2 (PAD)

LNA H

Mixer

PD_LNA_RFE_2 (0x010C[7])
PD_RLOOPB_1_RFE_2 (0x010C[6])
PD_RLOOPB_2_RFE_2 (0x010C[5])
PD_MXLOBUF_RFE_2 (0x010C[4])
PD_TIA_RFE_2 (0x010C[1])
ICT_LOOPB_RFE_2 (0x010F[14:10])
ICT_TIAMAIN_RFE_2 (0x010F[9:5])
ICT_TIAOUT_RFE_2 (0x010F[4:0])
ICT_LNACMO_RFE_2 (0x0110[14:10])
ICT_LNA_RFE_2 (0x0110[9:5])
ICT_LODC_RFE_2 (0x0110[4:0])

SEL_PATH_RFE_2 (0x010D[8:7])
PD_RLOOPB_1_RFE_2 (0x010C[6])
PD_RLOOPB_2_RFE_2 (0x010C[5])
EN_NEXTRX_RFE_2 (0x010D[0])

LO buff

SEL_PATH_RFE_2 (0x010D[8:7])
PD_LNA_RFE_2 (0x010C[7])
G_LNA_RFE_2 (0x0113[9:6])
EN_INSHSW_H_RFE_2 (0x010D[5])
CGSIN_LNA_RFE_2 (0x0111[4:0])

CDC_I_RFE_2 (0x010C[15:12])
CDC_Q_RFE_2 (0x010C[11:8])
SEL_PATH_RFE_2 (0x010C[8:7])
PD_QGEN_RFE_2 (0x010C[3])

SEL_PATH_RFE_2 (0x010D[8:7])
PD_MXLOBUF_RFE_2 (0x010C[4])

PD_RLOOPB_2_RFE_2 (0x010C[5])
G_RXLOOPB_RFE_2 (0x0113[5:2])
EN_INSHSW_LB2_RFE_2 (0x010D[3])

LO I/Q generator

RFB_TIA_RFE_2 (0x0114[4:0])
RCOMP_TIA_RFE_2 (0x0114[8:5])
PD_TIA_RFE_2 (0x010C[1])
G_TIA_RFE_2 (0x0113[1:0])
EN_DCOFF_RXFE_RFE_1(0x010D[6])
DCOFFI_RFE_2 (0x010E[13:7])
CFB_TIA_RFE_2 (0x0112[11:0])
CCOMP_TIA_RFE_2 (0x0112[15:12])

Frem: loopb2o(p/n)_trf2 (TX FE)

RX RF loopback Buffer

LO mixer buffer

TIA

To: rfeoi(p/n)_2 (RX BB)

From: rfg(p/n)_l_RFE_2 (PAD)
From: rfs(p/n)_l_RFE_2 (PAD)

LNA L

Mixer

SEL_PATH_RFE_2 (0x010D[8:7])
PD_LNA_RFE_2 (0x010C[7])
G_LNA_RFE_2 (0x0113[9:6])
EN_INSHSW_L_RFE_2 (0x010D[2])
CGSIN_LNA_RFE_2 (0x0111[4:0])

CDC_I_RFE_2 (0x010C[15:12])
CDC_Q_RFE_2 (0x010C[11:8])
SEL_PATH_RFE_2 (0x010D[8:7])
PD_QGEN_RFE_2 (0x010C[3])

TX mixer output cap

TIA

To: rfeoq(p/n)_2 (RX BB)

SEL_PATH_RFE_2 (0x010D[8:7])
PD_RLOOPB_1_RFE_2 (0x010C[6])
EN_NEXTRX_RFE_2 (0x010D[0])

LO buff

RFB_TIA_RFE_2 (0x0114[4:0])
RCOMP_TIA_RFE_2 (0x0114[8:5])
PD_TIA_RFE_2 (0x010C[1])
G_TIA_RFE_2 (0x0113[1:0])
EN_DCOFF_RXFE_RFE_1(0x010D[6])
DCOFFQ_RFE_2 (0x010E[6:0])
CFB_TIA_RFE_2 (0x0112[11:0])
CCOMP_TIA_RFE_2 (0x0112[15:12])

PD_RLOOPB_1_RFE_2 (0x010C[6])
G_RXLOOPB_RFE_2 (0x0113[5:2])
EN_INSHSW_LB1_RFE_2 (0x010D[4])

LO I/Q generator

Frem: loopb1o(p/n)_trf2 (TX FE)

RX RF loopback Buffer

LO mixer buffer

SEL_PATH_RFE_2 (0x010D[8:7])
PD_MXLOBUF_RFE_2 (0x010C[4])

From: rfg(p/n)_w_RFE_2 (PAD)
From: rfs(p/n)_w_RFE_2 (PAD)

LNA W

Mixer

CAP_RXMXO_RFE_2 (0x0111[9:5])

SEL_PATH_RFE_2 (0x010D[8:7])
PD_LNA_RFE_2 (0x010C[7])
G_LNA_RFE_2 (0x0113[9:6])
EN_INSHSW_W_RFE_2 (0x010D[1])
CGSIN_LNA_RFE_2 (0x0111[4:0])

RSSi

To: out_rssi_rfe_2 (AFE mux, or to PAD via tsdo<1> pin)

To: vrefout_rssi_rfe_2 (AFE mux, or to PAD via tstao pin)

EN_NEXTRX_RFE_2 (0x010D[0])

LO buff

PD_RSSI_RFE_2 (0x010C[2])

Not routed

Signal input/output
Analog signal lines
RESET_N (XXXX)   IO control name (register adress)

Figure 6 RFE2 control structure

72

A2.2 RBB Control Diagrams

Figure 7 RBB1 control structure

Figure 8 RBB2 control structure

A2.3 TRF Control Diagrams

Figure 9 TRF1 control structure

Figure 10 TRF1 control structure

A2.4 TBB Control Diagrams
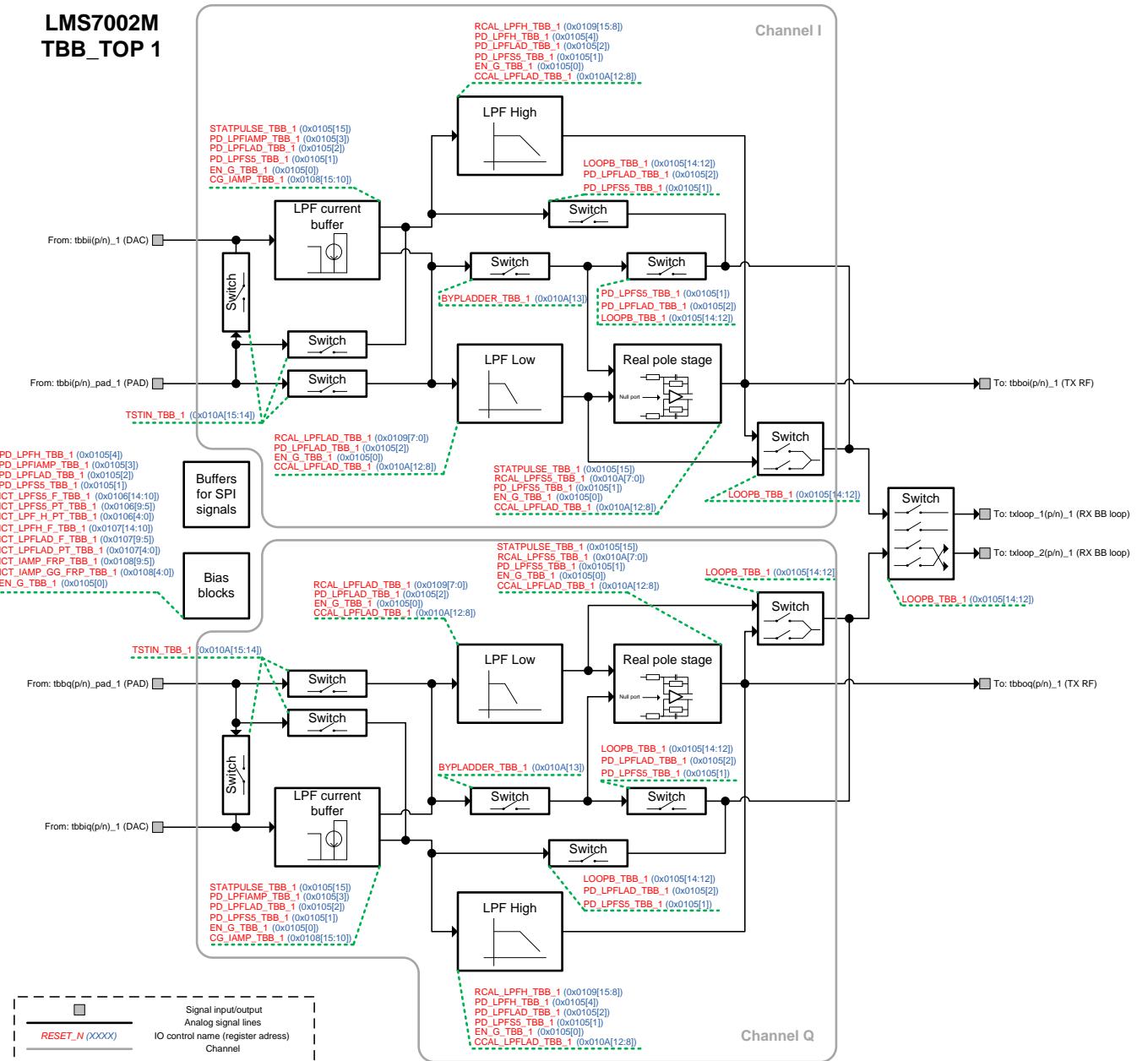
Figure 11 TBB1 control structure
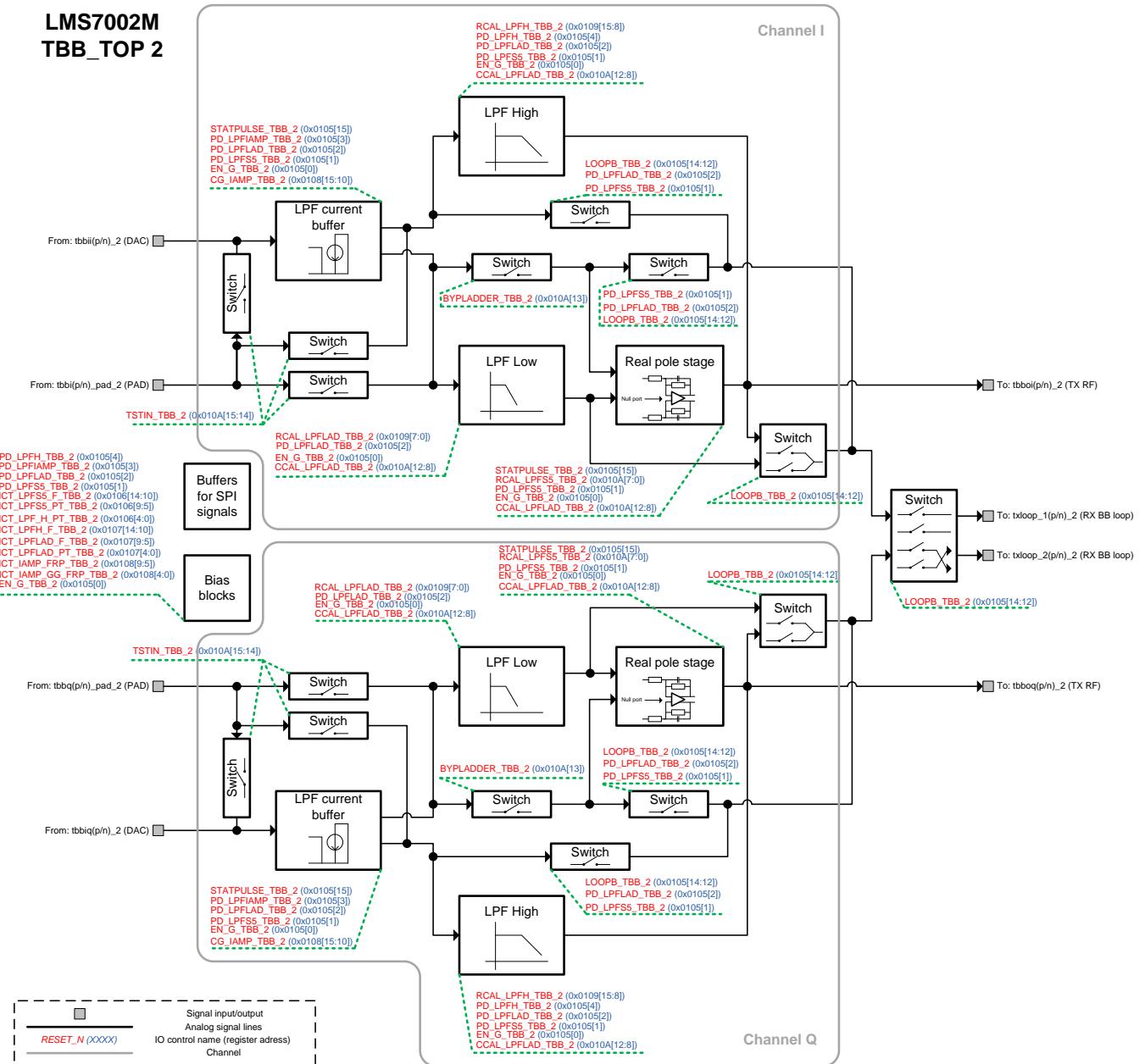
Figure 12 TBB2 control structure

A2.5 AFE Control Diagram
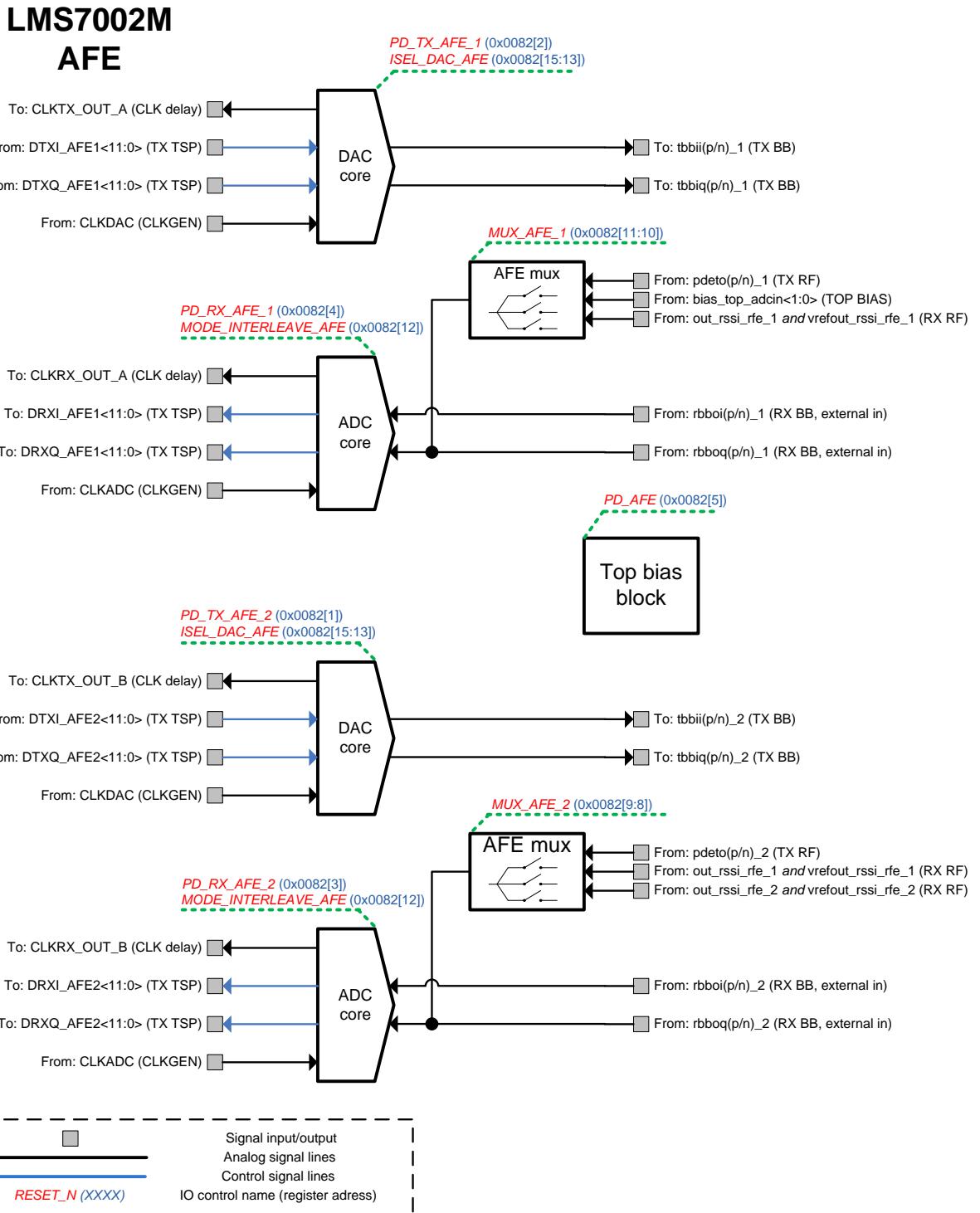
*Figure 13 AFE control structure*

A2.6 BIAS Control Diagram

LMS7002M
BIAS

PD_BIAS_MASTER (0x0084[0])

Bias enable block

From: CORE_LDO_EN (PAD)

AFE bias → To: ADC and DAC (AFE)

RP_CALIB_BIAS (0x0084[10:6])
PD_F_BIAS (0x0084[3])

From: CORE_LDO_EN (PAD)

Current Generator F → To: ip20f<47:1> (various blocks)
→ To: vr_rext (PAD)

PD_FRP_BIAS (0x0084[4])

From: CORE_LDO_EN (PAD)

Current Generator FRP
vr_cal_ref 10k → To: ip20frp<47:1> (various blocks)

PD_PTRP_BIAS (0x0084[2])

Current Generator PTRP
vptat_600m 10k → To: ip20ptrp<37:1> (various blocks)

RP_CALIB_BIAS (0x0084[10:6])
PD_PT_BIAS (0x0084[1])

Current Generator PT → To: ip20pt<50:1> (various blocks)

MUX_BIAS_OUT (0x0084[12:11])

From: vr_rext (PAD) and vr_cal_ref (internal)
From: vptat_600m (internal) and vr_cal_ref (internal)

BIAS mux → To: bias_top_adcin<1:0> (AFE MUX)

Signal input/output
Analog signal lines
RESET_N (XXXX) — IO control name (register adress)

Figure 14 BIAS control structure

A2.7 SXR and SXT Control Diagrams
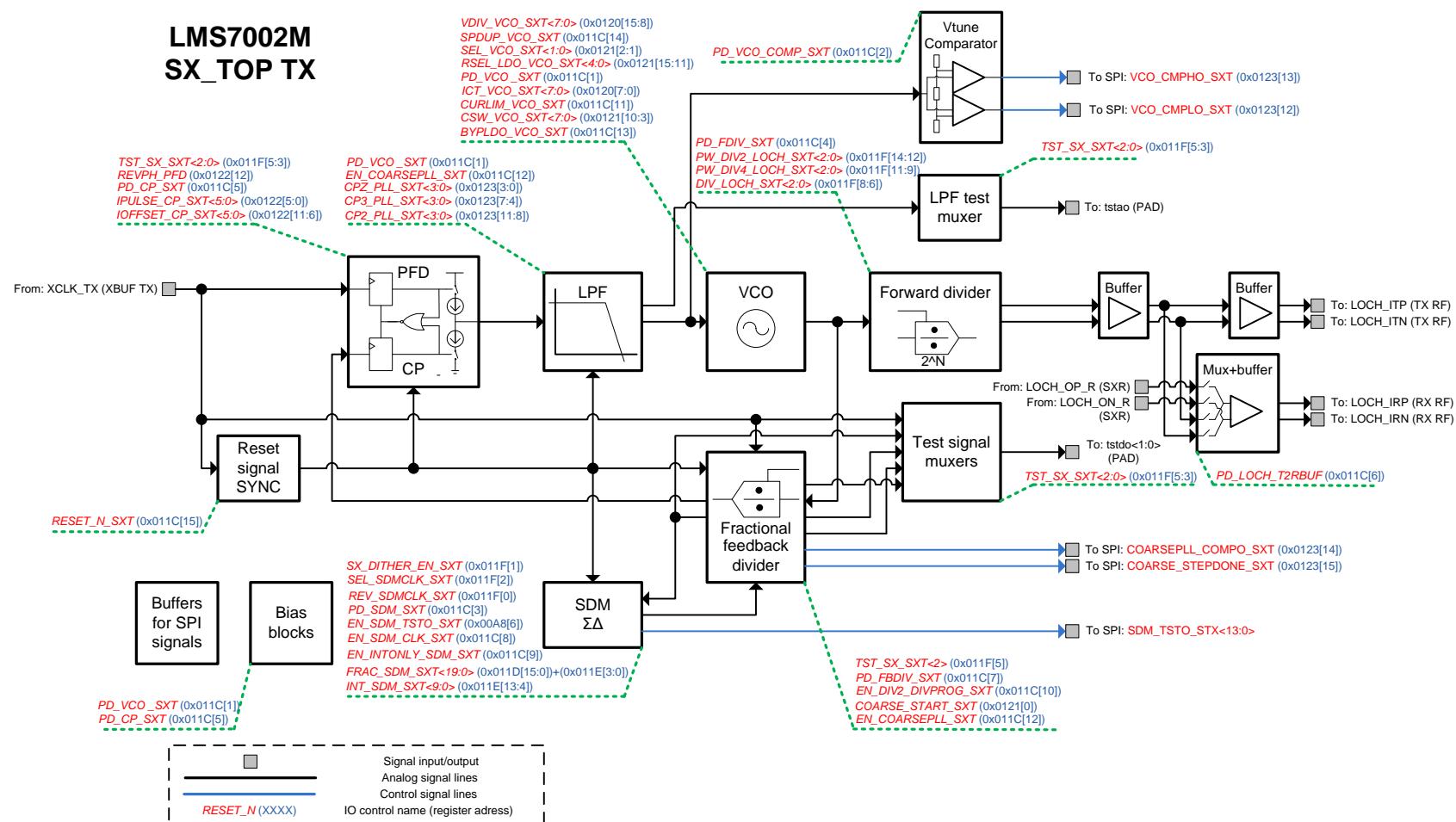
Figure 15 SXT control structure
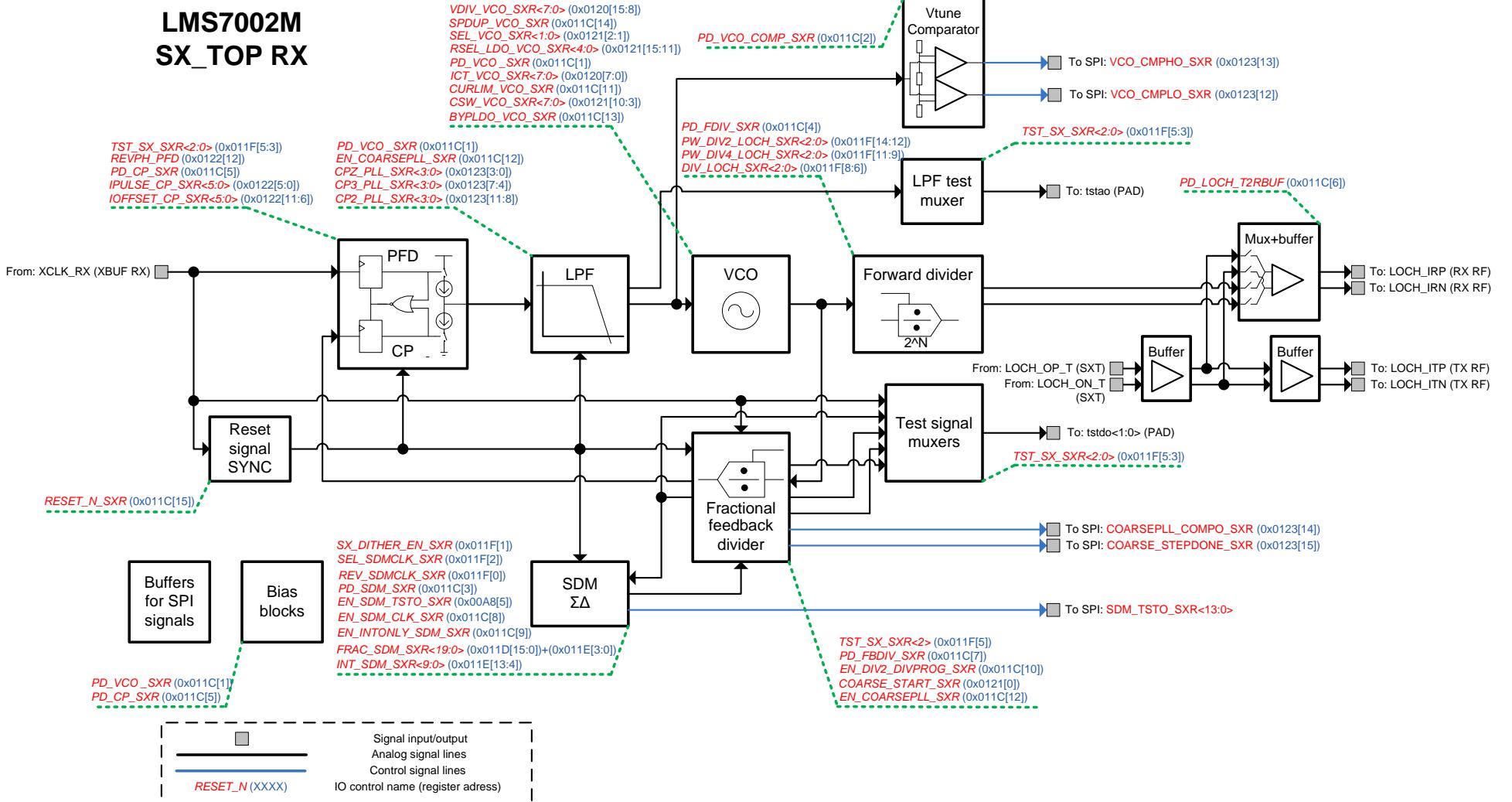
Figure 16 SXR control structure

A2.8 CGEN Control Diagram
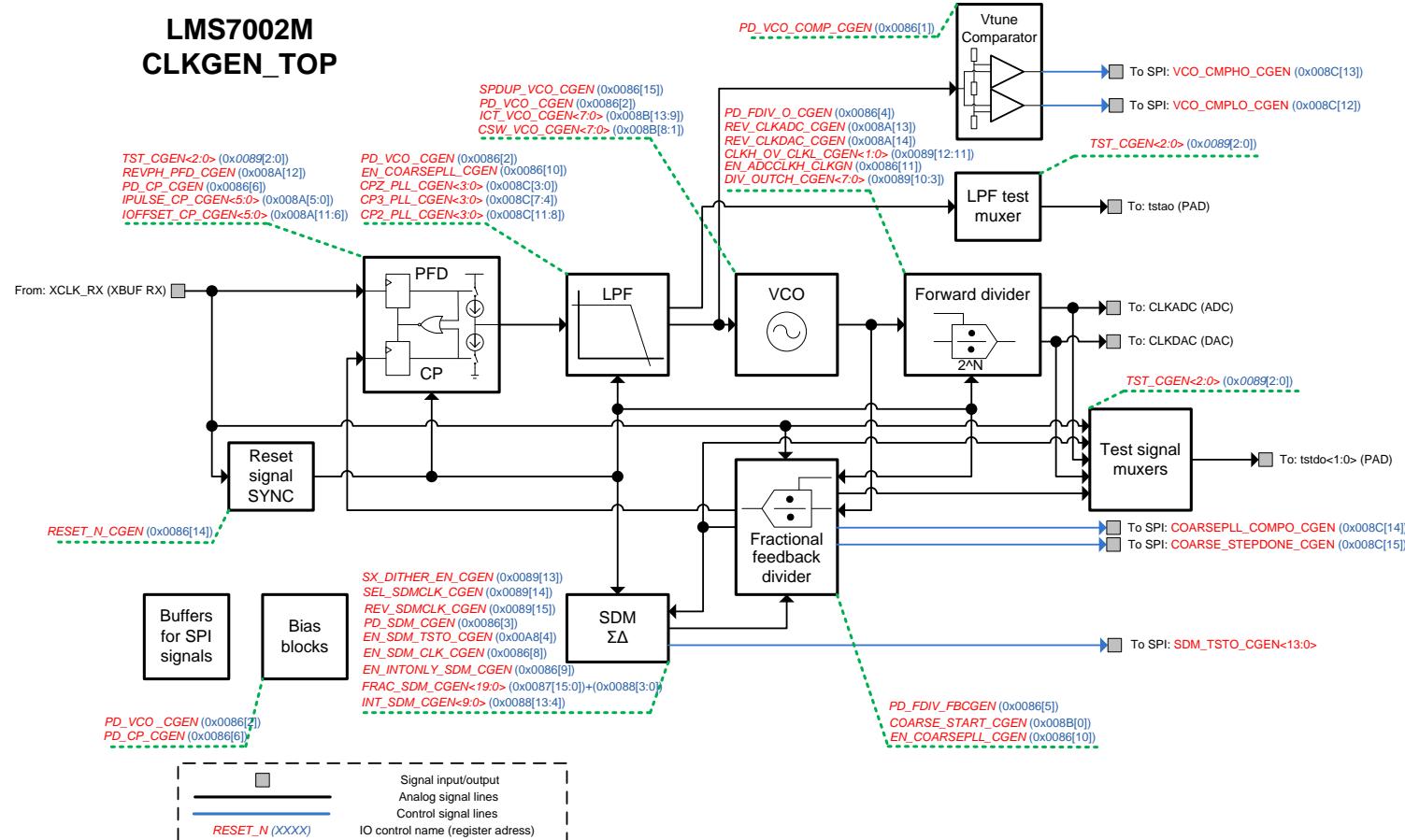
Figure 17 CGEN control structure

A2.9 XBUF Control Diagram
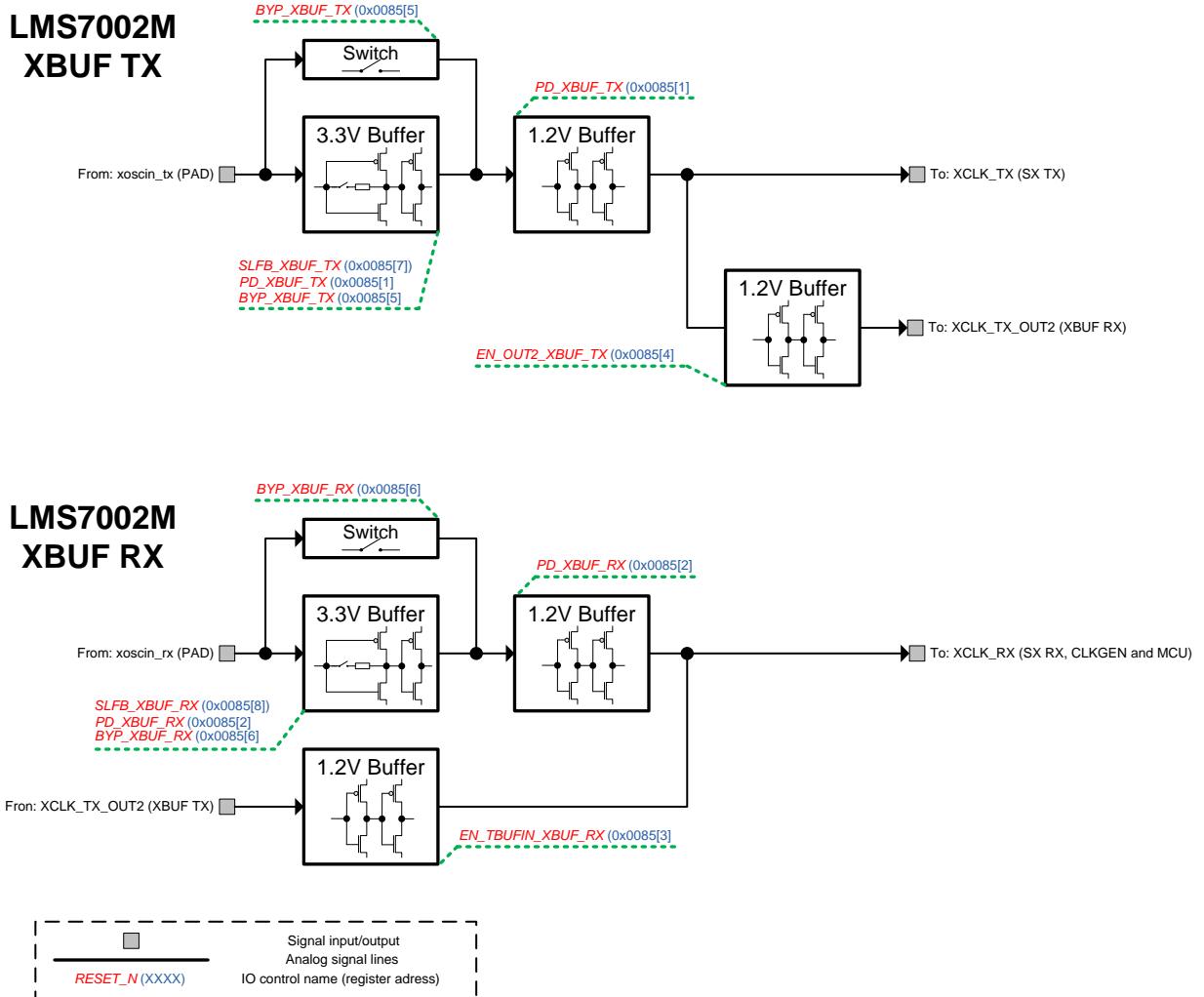
Figure 18 XBUF control structure

A2.10   LDOs Control Diagram
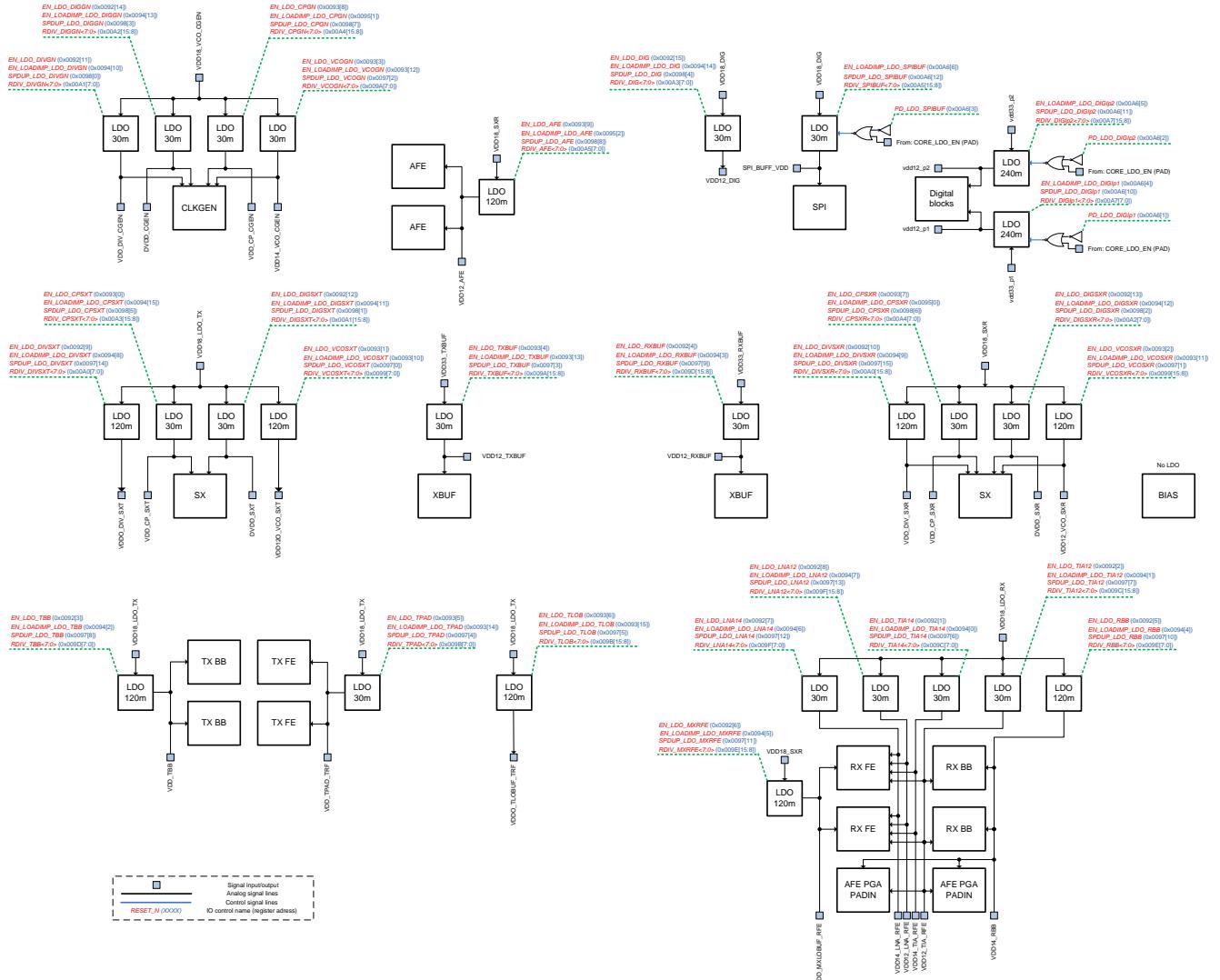
Figure 19 Control structure of LDOs

A2.11   CDS Control Diagram
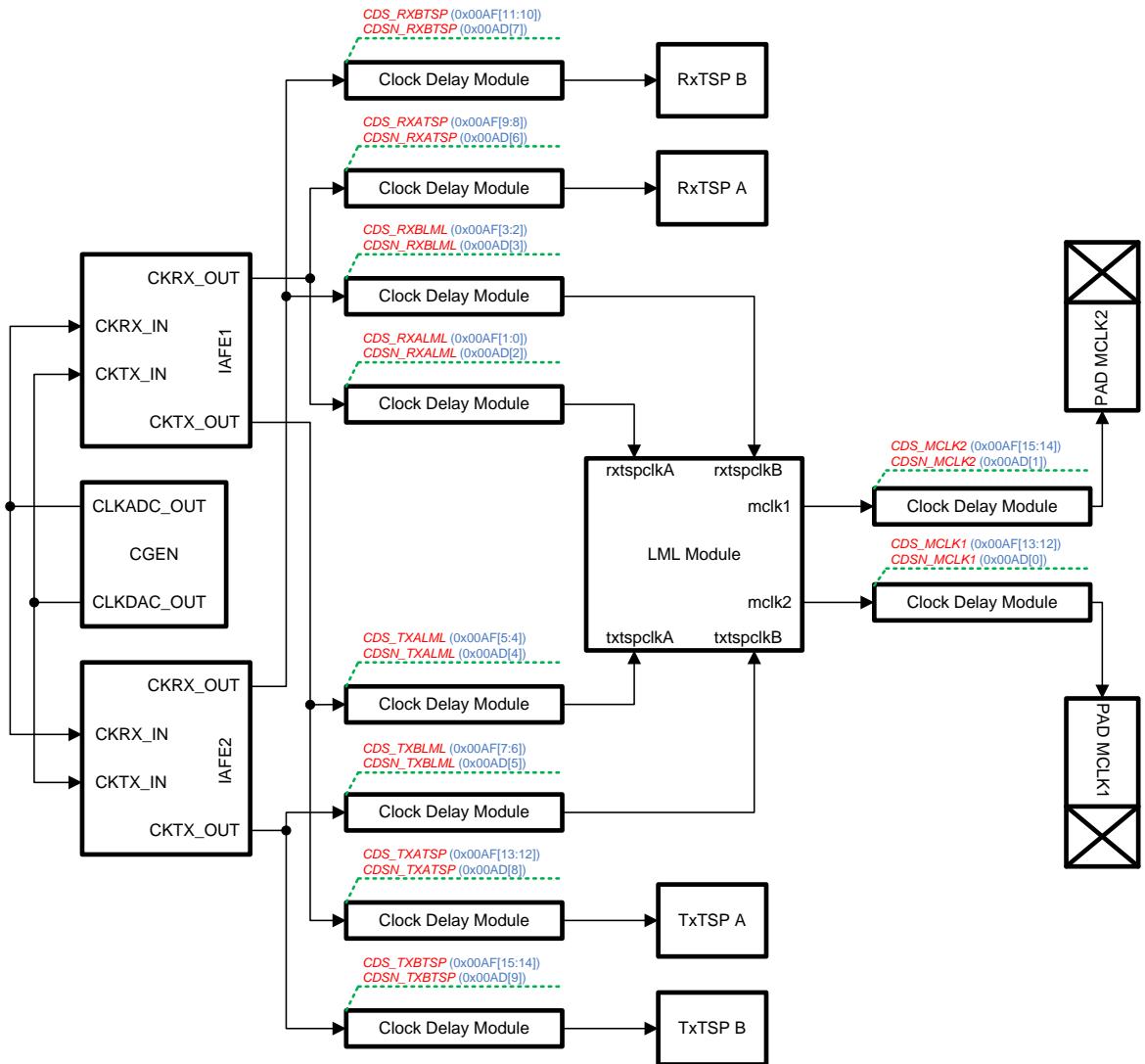
Figure 20 CDS control structure

A2.12   IO Cell Control Diagram
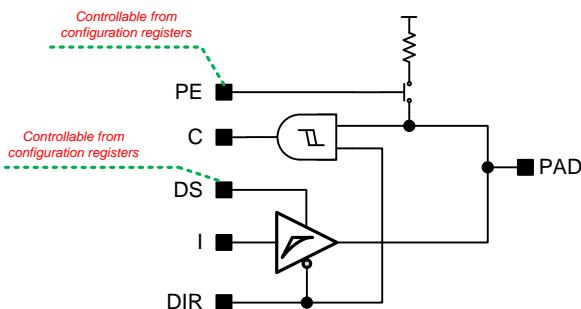
Figure 21 IO cell and controllable parameters

A2.13   TxTSP(A/B) Control Diagram
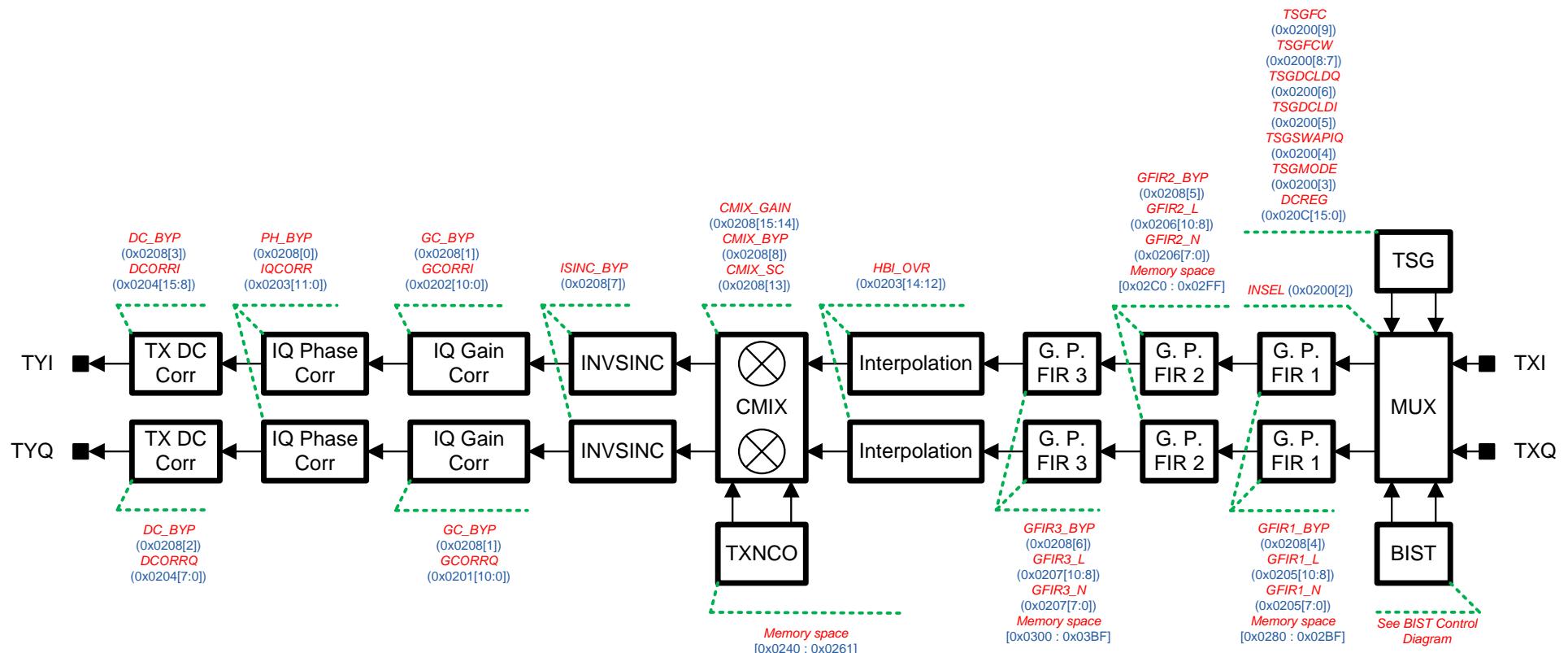
Figure 22 TxTSP(A/B) control structure

A2.14   RxTSP(A/B) Control Diagram
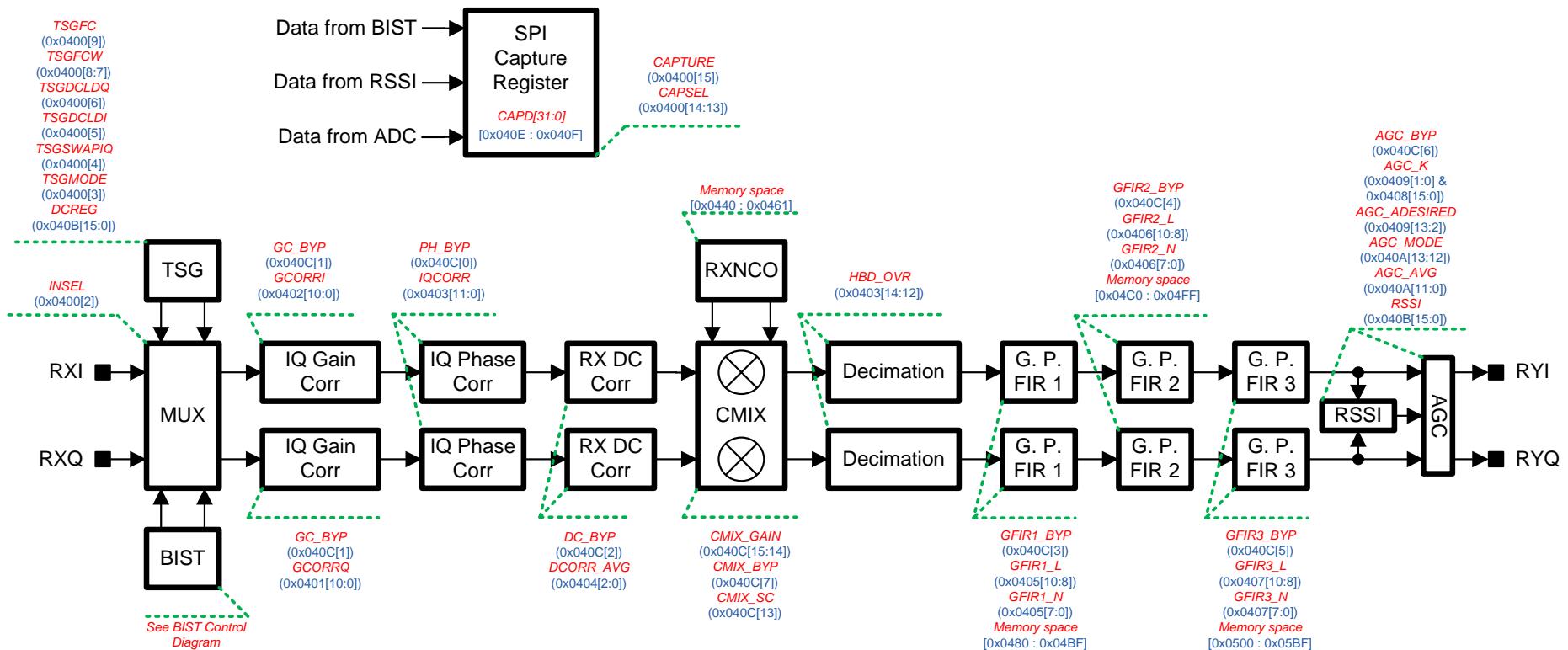
Figure 23 RxTSP(A/B) control structure

A2.15   SXR, SXT and CGEN BIST Control Diagram
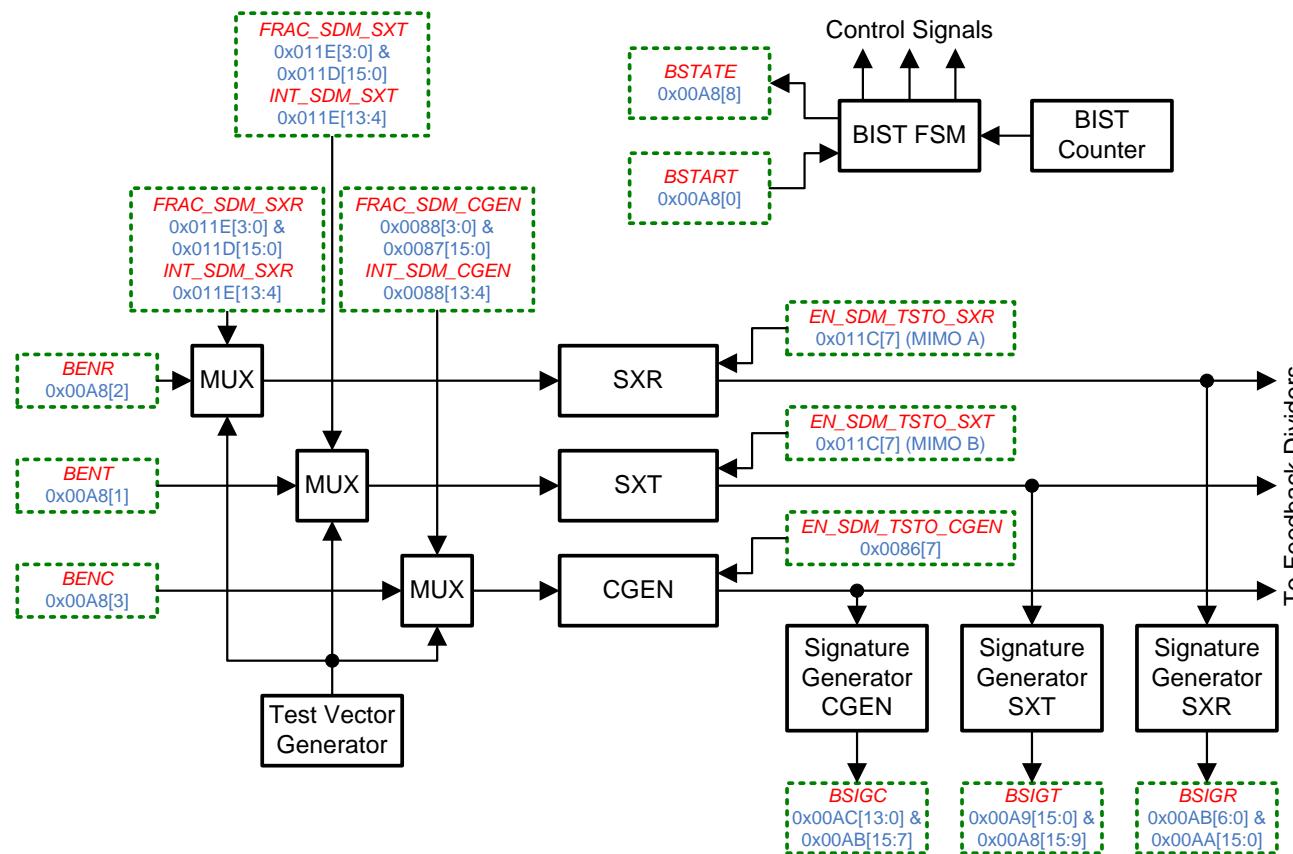
Figure 24 SXR, SXT and CGEN BIST control structure

A2.16   TxTSP(A/B) BIST Control Diagram
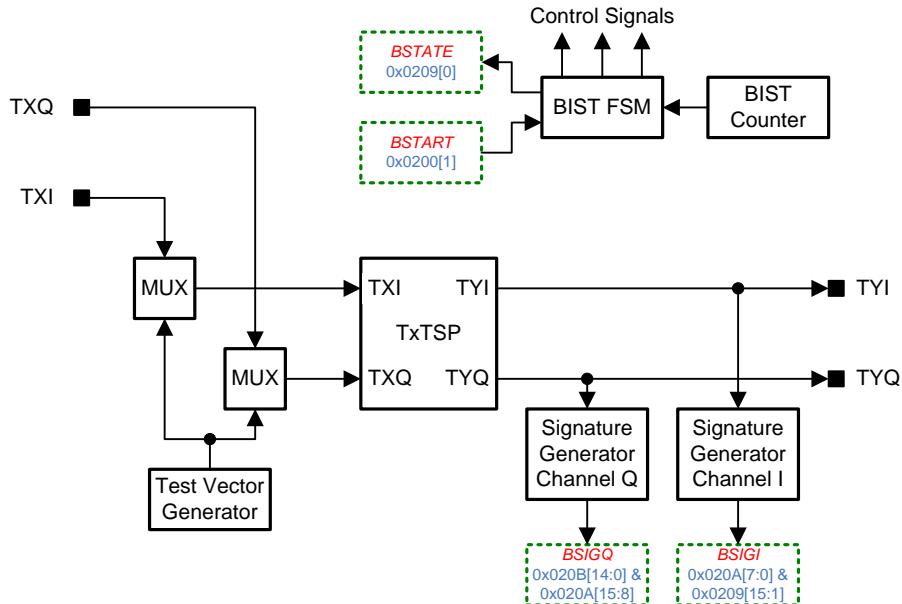
Figure 25 TxTSP(A/B) BIST control structure

A2.17   RxTSP(A/B) BIST Control Diagram
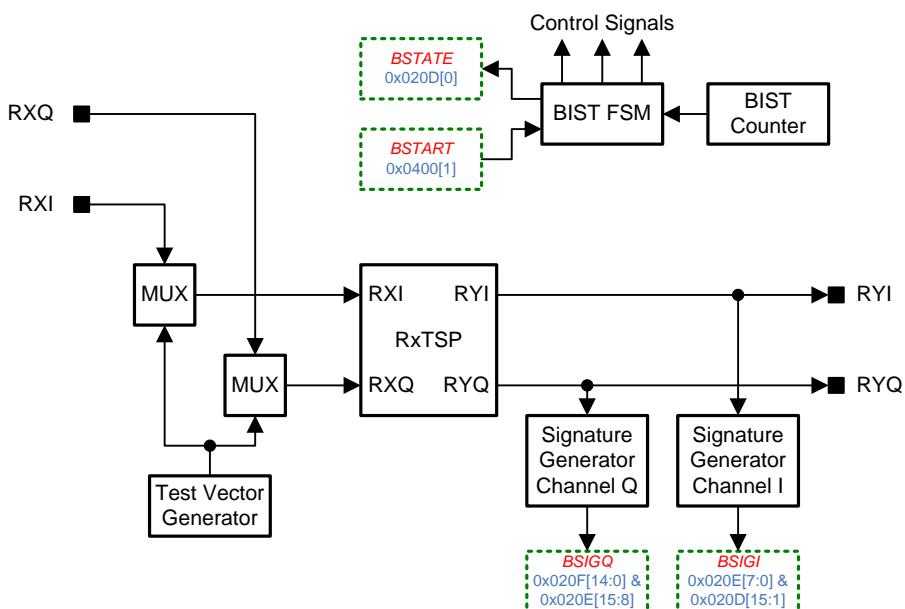
Figure 26 RxTSP(A/B) BIST control structure

A2.18 LimeLight™ Control Diagram
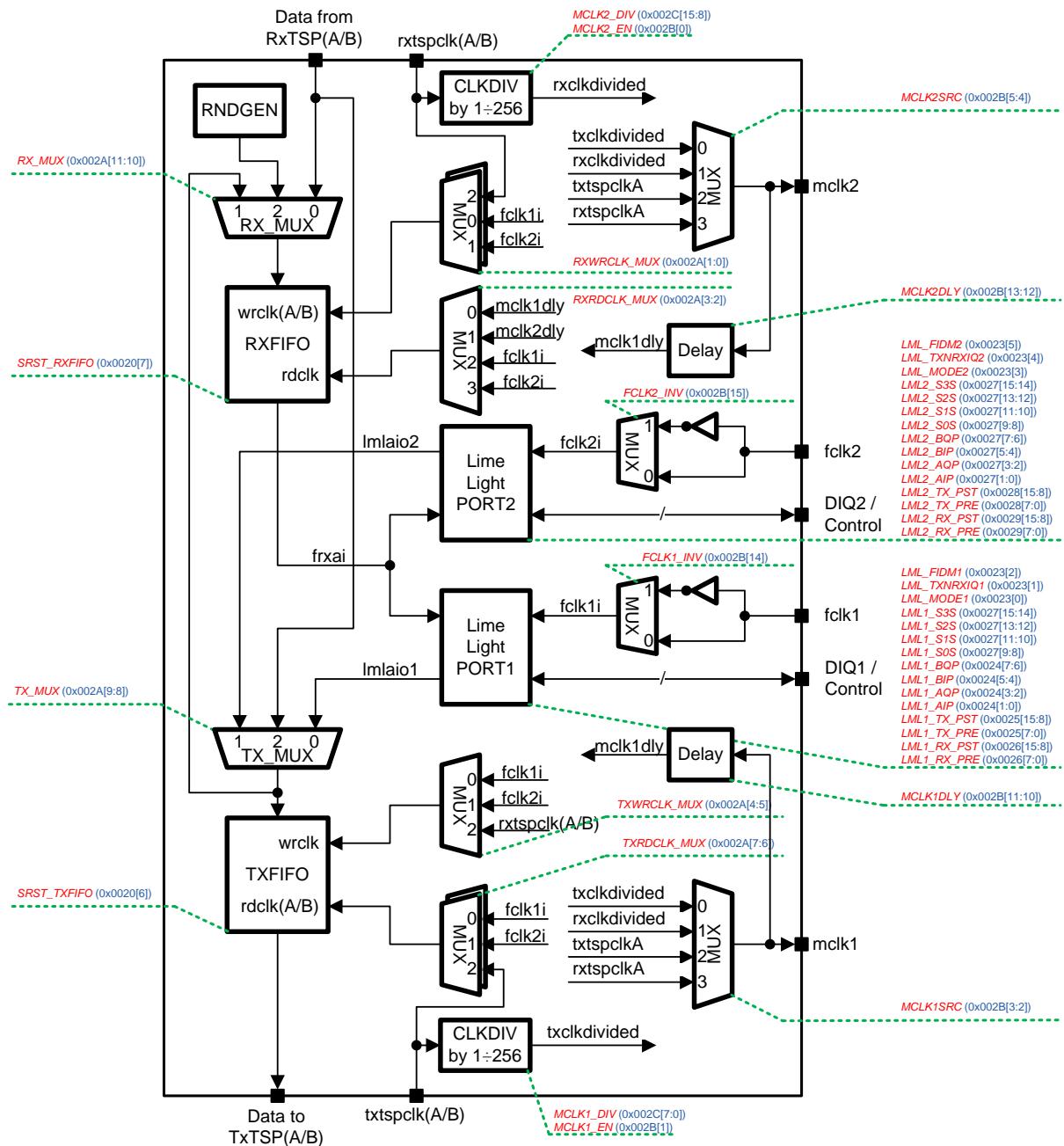
Figure 27 LimeLight™ control structure

Appendix 3

Calibration algorithms

A3.1 VCO coarse tuning

This chapter describes the algorithm for VCO coarse tuning, which finds the optimum SWC_VCO[7:0] value.

The VCO coarse tuning algorithm goes through the three following phases:
1. Initialization: sets the static control words for the synthesizer
2. Decision: monitors the two digital outputs from the synthesizer (COARSE_STEPDONE and COARSE_COMPO) and makes a decision on the correct value of every bit of SWC_VCO[7:0]. The Input control bit of COARSE_START will be used to start each cycle.
3. Normal mode setting: controls are set back to the values needed for normal synthesizer operation.

Below are two similar algorithms, one for the SXT/SXR and other CGEN. Algorithms differ mainly in the configuration register addresses.
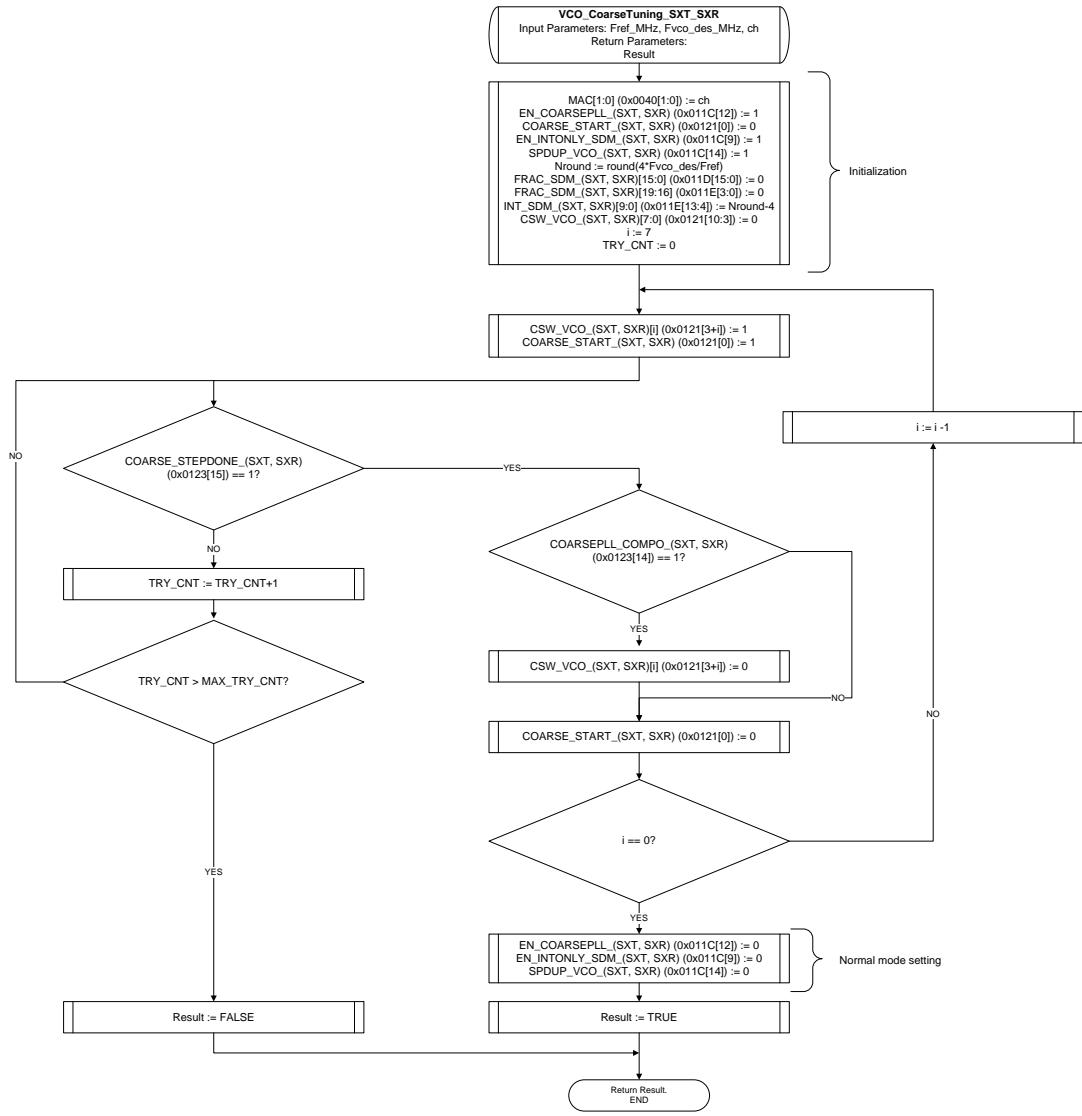
Figure 28 SXT SXR VCO Coarse tuning algorithm
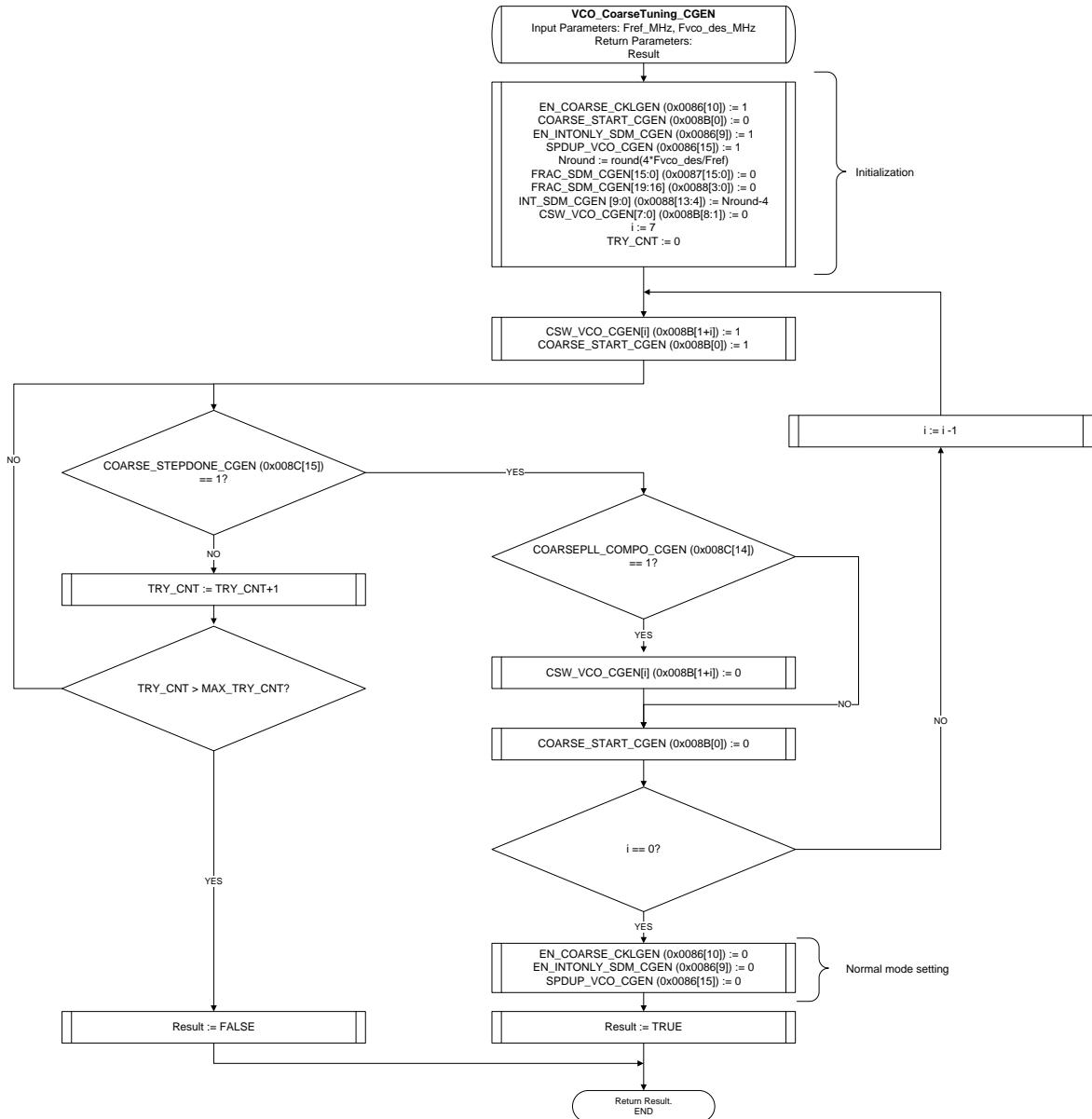
Figure 29 CGEN VCO Coarse tuning algorithm

C code for SXT/SXR VCO coarse tuning:

```
unsigned char VCO_CoarseTuning_SXT_SXR (float Fref_MHz, float Fvco_des_MHz, unsigned char ch)
{
        unsigned short Nround;
        unsigned char i, try_cnt;

        MIMO_Ctrl(ch);//SXT SXR selection

        // Initialization
        Modify_SPI_Reg_bits (0x011C, 12, 12, 1); // 1) EN_COARSEPLL=1, a. VCO control voltage is switched to a DC =VDD/2
        Modify_SPI_Reg_bits (0x0121, 0, 0, 0); // 2) COARSE_START=0
        Modify_SPI_Reg_bits (0x011C, 9, 9, 1); // 3) EN_INTONLY_SDM=1
        Modify_SPI_Reg_bits (0x011C, 14, 14, 1); // 4) SHORT_NOISEFIL=1 SPDUP_VCO_ Short the noise filter resistor to speed up the settling time
        Nround = (unsigned short)(4*Fvco_des_MHz/Fref_MHz+0.5); // 5) Nround=round(4*Fvco_des/Fref)
        Modify_SPI_Reg_bits (0x011D, 15, 0, 0); // 6) FRAC_SDM=0
        Modify_SPI_Reg_bits (0x011E, 3, 0, 0);
        Modify_SPI_Reg_bits (0x011E, 13, 4, (Nround-4)); // 7) INT_SDM=Nround-4
        Modify_SPI_Reg_bits (0x0121, 10, 3, 0); // 9)        Set SWC_VCO<7:0>=<00000000>
        i=7;// 10)    i=7

        //Decision - Loop Section
        while(1)
        {
                Modify_SPI_Reg_bits (0x0121, 3 + i, 3 + i, 1); // SWC_VCO<i>=1
                Modify_SPI_Reg_bits (0x0121, 0, 0, 1); // COARSE_START=1
```

```
                    while (Get_SPI_Reg_bits(0x0123, 15, 15) != 1) //wait till COARSE_STEPDONE=1
                    {
                            try_cnt++;
                            if(try_cnt > MAX_TRY_CNT) return 0;
                    }

                    if (Get_SPI_Reg_bits(0x0123, 14, 14) == 1) //check CAORSEPLL_COMPO
                    {
                            Modify_SPI_Reg_bits (0x0121, 3 + i, 3 + i, 0); // SWC_VCO<i>=0
                    }

                    Modify_SPI_Reg_bits (0x0121, 0, 0, 0); // COARSE_START=0

                    if(i==0) break;
                    i--;
            }

            //Normal mode setting
            Modify_SPI_Reg_bits (0x011C, 12, 12, 0); // EN_COARSEPLL=0
            Modify_SPI_Reg_bits (0x011C, 9, 9, 0); // EN_INTONLY_SDM=0
            Modify_SPI_Reg_bits (0x011C, 14, 14, 0); // SHORT_NOISEFIL=0 SPDUP_VCO_ Short the noise filter resistor to speed up the settling time

            return 1;
}
```

C code for CGEN VCO coarse tuning:

```
unsigned char VCO_CoarseTuning_CGEN (float Fref_MHz, float Fvco_des_MHz)
{
            unsigned short Nround;
            unsigned char i, try_cnt;

            // Initialization
            Modify_SPI_Reg_bits (0x0086, 10, 10, 1); // 1) EN_COARSE_CKLGEN=1, a. VCO control voltage is switched to a DC =VDD/2
            Modify_SPI_Reg_bits (0x008B, 0, 0, 0); // 2) COARSE_START_CGEN=0
            Modify_SPI_Reg_bits (0x0086, 9, 9, 1); // 3) EN_INTONLY_SDM_CGEN=1
            Modify_SPI_Reg_bits (0x0086, 15, 15, 1); // 4) SHORT_NOISEFIL=1 SPDUP_VCO_CGEN Short the noise filter resistor to speed up the settling time
            Nround = (unsigned short)(4*Fvco_des_MHz/Fref_MHz+0.5); // 5) Nround=round(4*Fvco_des/Fref)
            Modify_SPI_Reg_bits (0x0087, 15, 0, 0); // 6) FRAC_SDM_CGEN=0
            Modify_SPI_Reg_bits (0x0088, 3, 0, 0);
            Modify_SPI_Reg_bits (0x0088, 13, 4, (Nround-4)); // 7) INT_SDM_CGEN =Nround-4
            Modify_SPI_Reg_bits (0x008B, 8, 1, 0); // 9)        Set CSW_VCO_CGEN<7:0>=<00000000>
            i=7;// 10)    i=7

            // Loop Section
            while(1)
            {
                    Modify_SPI_Reg_bits (0x008B, 1 + i, 1 + i, 1); // CSW_VCO_CGEN<i>=1
                    Modify_SPI_Reg_bits (0x008B, 0, 0, 1); // COARSE_START_CGEN=1

                    while ( Get_SPI_Reg_bits(0x008C, 15, 15) != 1 ) //wait till COARSE_STEPDONE_CGEN=1
                    {
                            try_cnt++;
                            if(try_cnt > MAX_TRY_CNT) return 0;
                    }

                    if ( Get_SPI_Reg_bits(0x008C, 14, 14) == 1 ) //check COARSEPLL_COMPO_CGEN
                    {
                            Modify_SPI_Reg_bits (0x008B, 1 + i, 1 + i, 0); // SWC_VCO<i>=0
                    }

                    Modify_SPI_Reg_bits (0x008B, 0, 0, 0); // 2) COARSE_START_CGEN=0

                    if(i==0) break;
                    i--;
            }

            Modify_SPI_Reg_bits (0x0086, 10, 10, 0); // 1) EN_COARSE_CKLGEN=0
            Modify_SPI_Reg_bits (0x0086, 9, 9, 0); // 3) EN_INTONLY_SDM_CGEN=0
            Modify_SPI_Reg_bits (0x0086, 15, 15, 0); // 4) SHORT_NOISEFIL=0 SPDUP_VCO_CGEN Short the noise filter resistor to speed up the settling time

            return 1;
}
```

A3.2 Main resistor (bias) calibration

Calibration steps:

1. Set the control signal MUX_BIAS_OUT=1
2. Sweep from zero to maximum RP_CALIB_BIAS; in each step:

- Use the Q input of the ADC of Channel 1 to read the difference between on the chip fixed voltage and the off-chip voltage.
- Compare the ADC value with the best value (which is initially set to very high). If the ADC value is lower, then save it as "Best Value".
3. Return the "Best Value" found during the sweep.
4. Calculate and return ratio of "Best Value" to the "Nominal Value". This ratio is the calibration value of the resistor for other calibration algorithms to use.
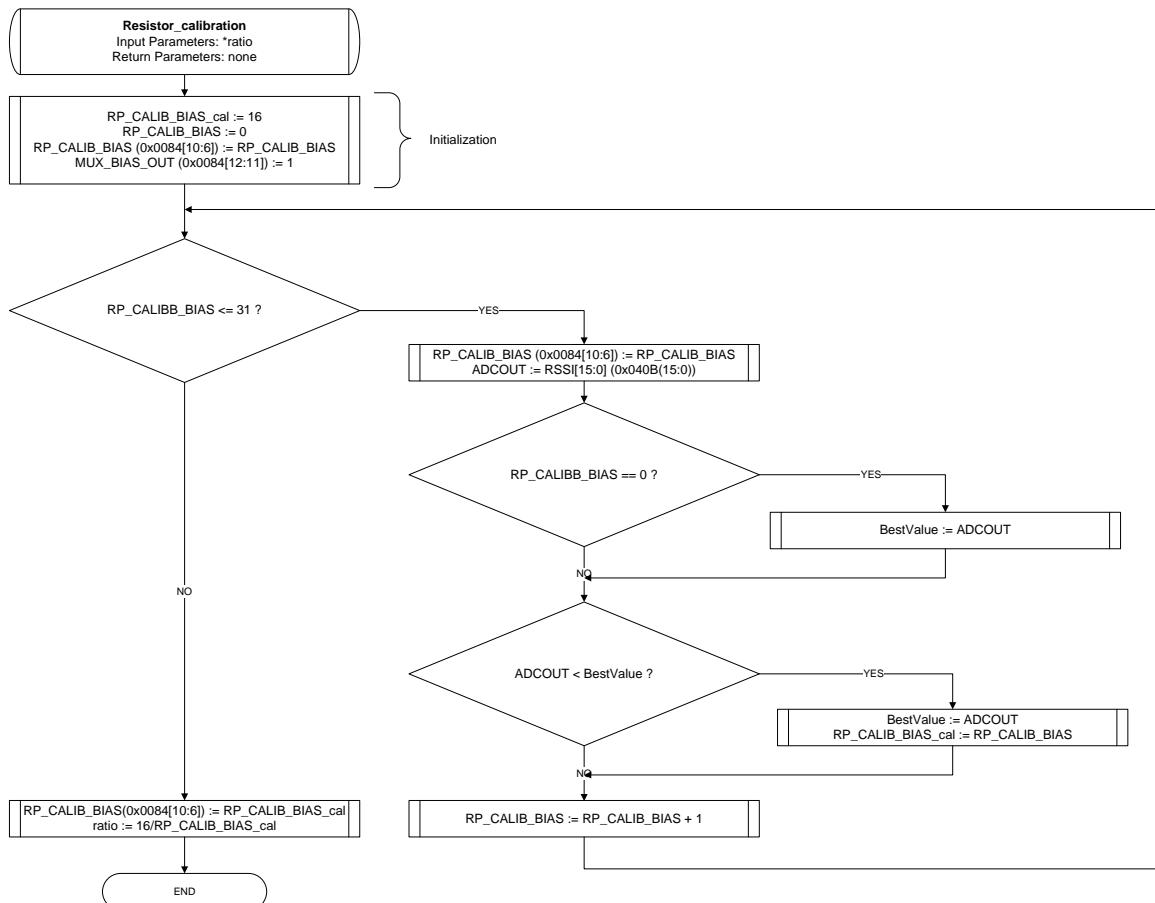

Figure 30 Resistor calibration algorithm

The following is the C code that implements the described algorithm:

```
void Resistor_calibration (float *ratio)
{
        unsigned char RP_CALIB_BIAS, RP_CALIB_BIAS_cal;
        unsigned short BestValue, ADCOUT;

        RP_CALIB_BIAS_cal = 16;
        RP_CALIB_BIAS = 0;
        Modify_SPI_Reg_bits (0x0084, 10, 6, RP_CALIB_BIAS); // write RP_CALIB_BIAS value
        Modify_SPI_Reg_bits (0x0084, 12, 11, 1); // MUX_BIAS_OUT = 1

        while (RP_CALIB_BIAS <= 31)
        {
                Modify_SPI_Reg_bits (0x0084, 10, 6, RP_CALIB_BIAS); // write RP_CALIB_BIAS value
                ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value

                if(RP_CALIB_BIAS == 0)
                {
                        BestValue = ADCOUT;
                }

                if ( ADCOUT < BestValue )
                {
                        BestValue = ADCOUT;
                        RP_CALIB_BIAS_cal = RP_CALIB_BIAS; //store calibrated value
                }
```

```
                RP_CALIB_BIAS++;
        }

        Modify_SPI_Reg_bits (0x0084, 10, 6, RP_CALIB_BIAS_cal); // set the control RP_CAL_BIAS to stored calibrated value
        *ratio = (float) 16/RP_CALIB_BIAS_cal; //calculate ratio
}
```

A3.3 RBB calibration

RBB calibration is divided into two calibrations for low and high bands. Each calibration consists of several smaller algorithms.

A3.3.1   RBB Low Band Calibration

Calibration steps:

1. Save the current configuration
2. Start with calibrated value of the R.
3. Approximate (by calculation) the control value of the RBANK.=>Register the value of the RBB RBANKs control.
4. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 1.4 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 1.4 MHz rxMode.
5. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 3 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 3 MHz rxMode.
6. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 5 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 5 MHz rxMode.
7. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 10 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 10 MHz rxMode.
8. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 15 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 15 MHz rxMode.
9. Calibrate (by measurement using loopback path 7) the control value of the CBANK(Low Band Section) at the 20 MHz/2 bandwidth. => Register the CBANK control value for the low-band section for 20 MHz rxMode.
10. Restore the configuration
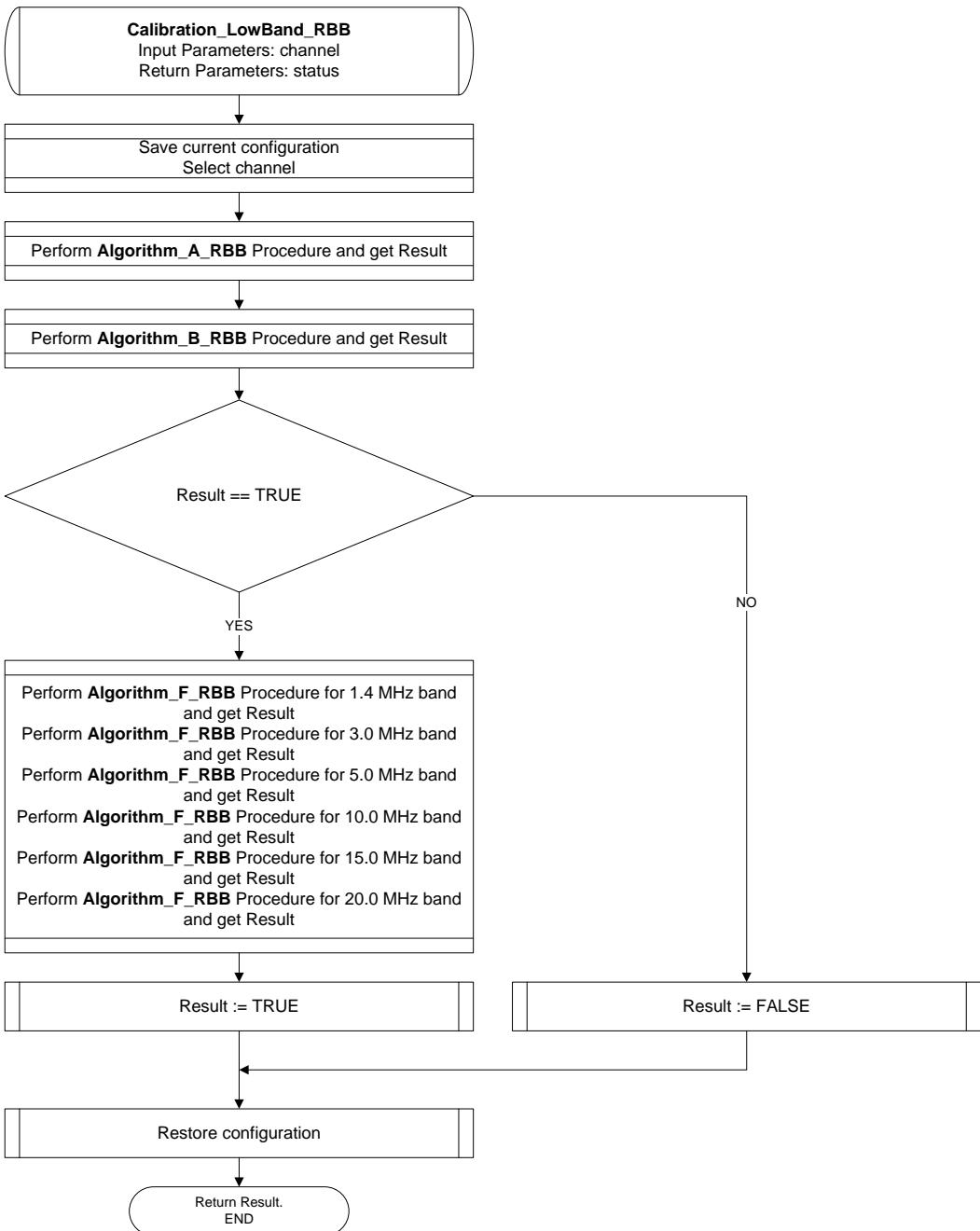
Figure 31 RBB Low Band calibration algorithm

The following is the C code that implements the described algorithm:

```
unsigned char Calibration_LowBand_RBB (unsigned char ch)
{
        unsigned char result = 0;

        Save_config_RBB (); //save current configuration

        MIMO_Ctrl (ch);
        Modify_SPI_Reg_bits (0x040A, 13, 12, 1); // AGC Mode = 1 (RSSI mode);

        Algorithm_A_RBB (); // Aproximate resistor value for RBB RBANKS (Algorithm A)

        Set_cal_path_RBB (7); // Set control signals to path 7 (RX LowBand)

        if (Algorithm_B_RBB (&LowFreqAmp) != 1) goto RESTORE; // Calibrate and Record the low frequency output amplitude (Algorithm B)

        Algorithm_F_RBB (RBB_1_4MHZ);// CalibrateByCap the output cuttoff frequency at 0,7 MHz and store
        Algorithm_F_RBB (RBB_3_0MHZ);// CalibrateByCap the output cuttoff frequency at 1,5 MHz and store
        Algorithm_F_RBB (RBB_5_0MHZ);// CalibrateByCap the output cuttoff frequency at 2,5 MHz MHz and store
        Algorithm_F_RBB (RBB_10_0MHZ);// CalibrateByCap the output cuttoff frequency at 5 MHz MHz and store
        Algorithm_F_RBB (RBB_15_0MHZ);// CalibrateByCap the output cuttoff frequency at 7,5 MHz and store
        Algorithm_F_RBB (RBB_20_0MHZ);// CalibrateByCap the output cuttoff frequency at 10 MHz MHz and store
```

```
            result = 1;

        RESTORE:
        Restore_config_RBB ();      //restore configuration

            return result;

}
```

A3.3.2   RBB High band Calibration

Calibration steps:

1. Save the current configuration
2. Select channel
3. Calibrate (by measurement using loopback path 8) the control value of the CBANK(High Band Section) at the 37 MHz/2 bandwidth. => Register the CBANK control value for the high-band section for 37 MHz rxMode.
4. Calibrate (by measurement using loopback path 8) the control value of the CBANK(High Band Section) at the 66 MHz/2 bandwidth. => Register the CBANK control value for the high-band section for 66 MHz rxMode.
5. Calibrate (by measurement using loopback path 8) the control value of the CBANK(High Band Section) at the 108 MHz/2 bandwidth. => Register the CBANK control value for the high-band section for 108 MHz rxMode.
6. Restore the configuration
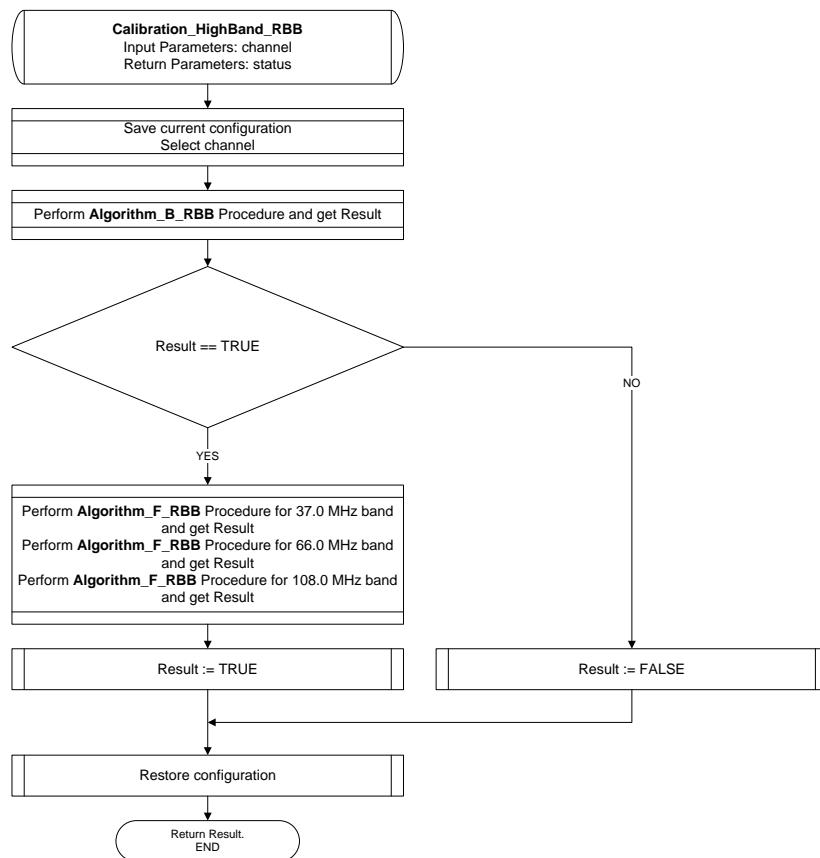
Figure 32 RBB High Band calibration algorithm

The following is the C code that implements the described algorithm:

```
unsigned char Calibration_HighBand_RBB (unsigned char ch)
{
        unsigned char result = 0;

        Save_config_RBB (); //save current configuration

        MIMO_Ctrl (ch);
        Modify_SPI_Reg_bits (0x040A, 13, 12, 1); // AGC Mode = 1 (RSSI mode)

        Set_cal_path_RBB (8); //Set control signals to path 8 (RX HighBand)

        if (Algorithm_B_RBB (&LowFreqAmp) != 1) goto RESTORE; // Calibrate and Record the low frequency output amplitude (Algorithm B)

        Algorithm_F_RBB (RBB_37_0MHZ);// CalibrateByCap the output cuttoff frequency at 18,5 MHz MHz and store
        Algorithm_F_RBB (RBB_66_0MHZ);// CalibrateByCap the output cuttoff frequency at 33 MHz MHz and store
        Algorithm_F_RBB (RBB_108_0MHZ);// CalibrateByCap the output cuttoff frequency at 54 MHz MHz and store

        RESTORE:
        Restore_config_RBB ();       //restore configuration

        return result;
}
```

A3.4 Nested algorithms

A3.4.1   Algorithm A

Multiply the ratio of the on-chip resistor to the off-chip resistor by the default control value (R_CTL_LPF_RBB) of the respective resistor.
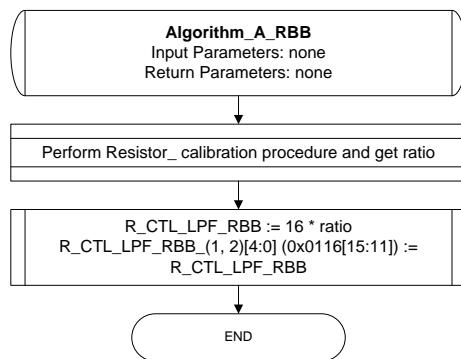

Figure 33 RBB algorithm A

C code for algorithm A:

```
void Algorithm_A_RBB ()
{
        unsigned char R_CTL_LPF_RBB;
        float ratio;

        Resistor_calibration (&ratio);
        R_CTL_LPF_RBB = (unsigned char)(16 * ratio); // Default control value multiply by ratio

        Modify_SPI_Reg_bits (0x0116, 15, 11, R_CTL_LPF_RBB);
        RBB_RBANK[MIMO_ch] = R_CTL_LPF_RBB; // Store RBANK Values (R_CTL_LPF_RBB)
}
```

A3.4.2 Algorithm B

Algorithm steps:

1. Set the DAC output to 100 kHz single tone.
2. Start with the nominal setting value for "CG_IAMP_TBB".
3. Linearly and proportionally adjust "CG_IAMP_TBB<5:0>" control lines to have about 80% of full scale swing. For this: measure the output, if the output was lower or higher than 80%, then adjust "CG_IAMP_TBB" proportionally and retest and measure for verification.
4. Record the exact value of the amplitude in "LowFreqAmp" for later on comparison.
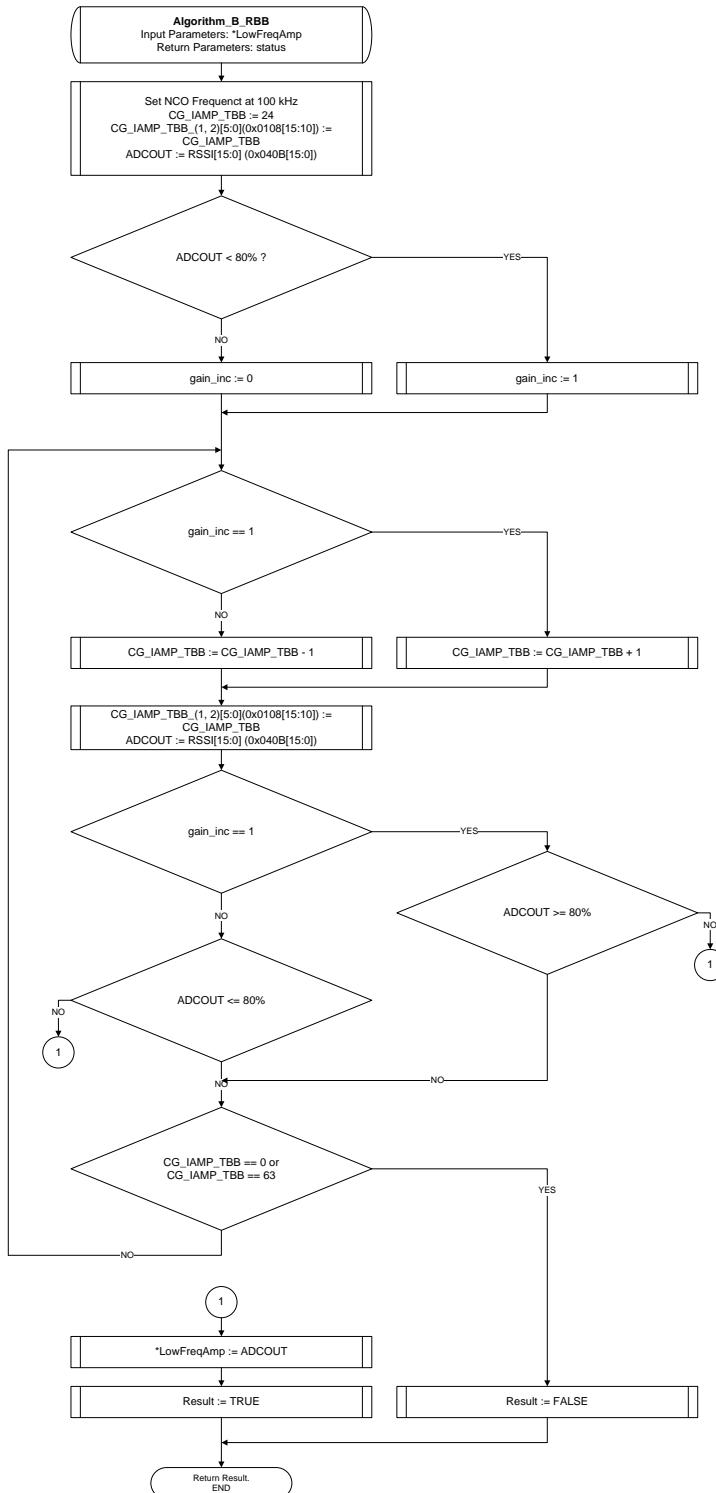
Figure 34 RBB algorithm B

C code for algorithm B:

```
unsigned char Algorithm_B_RBB (unsigned short *LowFreqAmp)
{
            unsigned short ADCOUT;
            unsigned char CG_IAMP_TBB, gain_inc;

            Set_NCO_Freq (0.1); // Set DAC output to 100kHz (0.1MHz) single tone.

            CG_IAMP_TBB = 24; //set nominal CG_IAMP_TBB value
            Modify_SPI_Reg_bits (0x0108, 15, 10, CG_IAMP_TBB); //write val to reg

            //Modify_SPI_Reg_bits (0x040A, 13, 12, 1); // AGC Mode = 1 (RSSI mode)
            ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the output level at the ADC input
```

```
if(ADCOUT < 52428) gain_inc = 1; //is it less then 80% of full scale swing (52428 (80% of 16 bits))
        else gain_inc = 0;

while (1)
{
        if(gain_inc) CG_IAMP_TBB++;
                else CG_IAMP_TBB--;

        Modify_SPI_Reg_bits (0x0108, 15, 10, CG_IAMP_TBB); //write val to reg

        ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the output level at the ADC input

        if (gain_inc)
        {
                if(ADCOUT >= 52428) break;
        }
        else
        {
                if(ADCOUT <= 52428) break;
        }

        if( (CG_IAMP_TBB == 0) || (CG_IAMP_TBB == 63))  //gain limit reached
        {
                return 0;
                break;
        }
}

*LowFreqAmp = ADCOUT;
return 1;
}
```

A3.4.3   Algorithm F

Algorithm steps:

1. If   ("CalFreq")   <=10   MHz,   then   CONTROL=C_CTL_LPFL_RBB,   else,
   CONTROL=C_CTL_LPFH_RBB
2. Set the CONTROL to maximum value. This should bring the output cut-off frequency
   to minimum.
3. Apply a single tone frequency at "CalFreq".
4. Measure the value of the amplitude at the ADC input.
5. If ADC value >= LowFreqAmp, then jump back to line #8.
6. Decrease the CONTROL value by one.
7. Jump back to line #4
8. Save the value of the CONTROL.
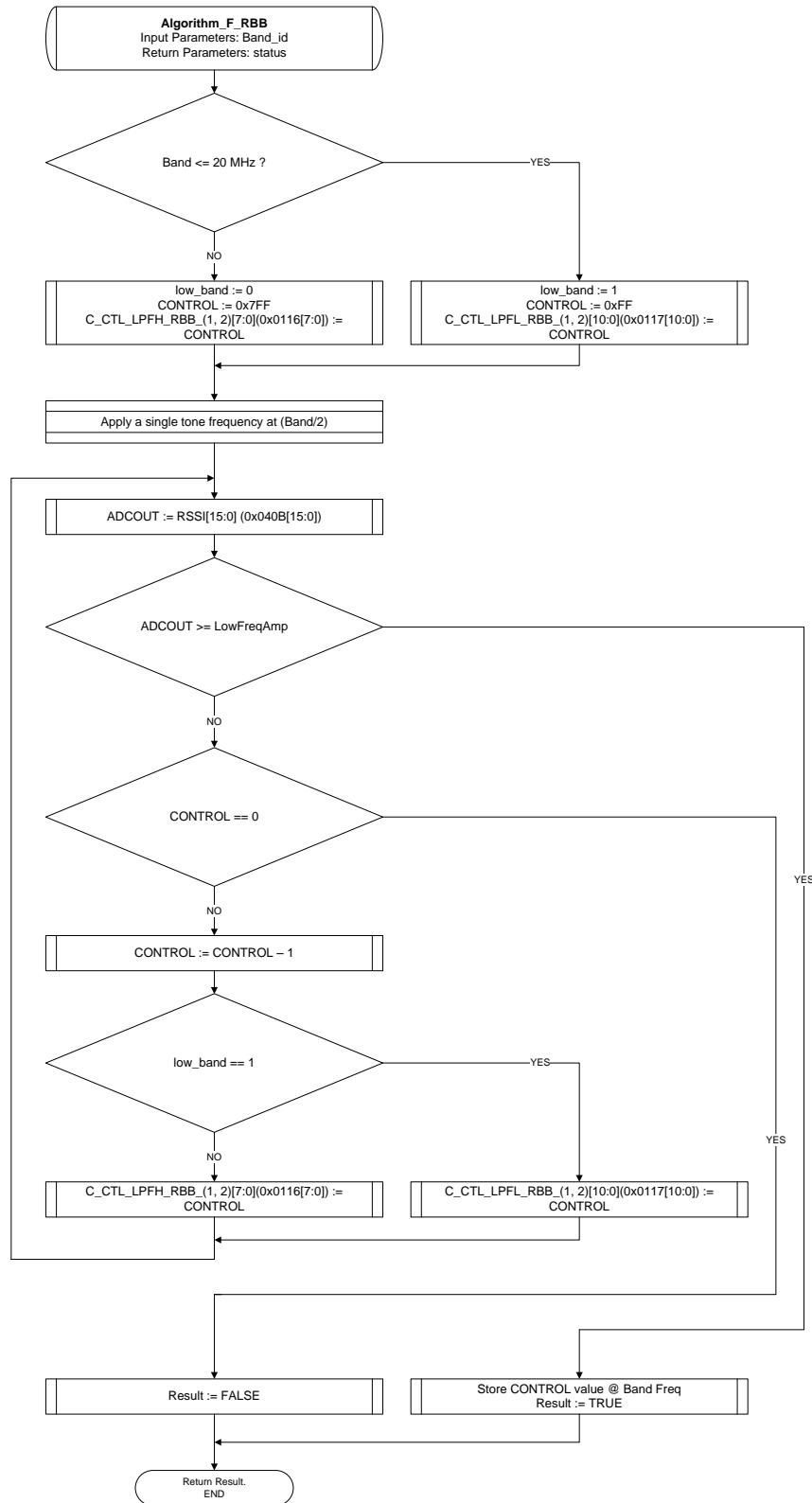
Figure 35 RBB algorithm F

C code for algorithm F:

```
unsigned char Algorithm_F_RBB (unsigned char Band_id)
{
        unsigned short ADCOUT, CONTROL;
        unsigned char low_band;

        //Modify_SPI_Reg_bits (0x040A, 13, 12, 1); // AGC Mode = 1 (RSSI mode)
```

```
if(Band_id <= RBB_20_0MHZ) //low band
{
         low_band = 1; // CONTROL=C_CTL_LPFL_RBB
         CONTROL = 0xFF; // Set the CONTROL to maximum value. This should bring the output cutt-off frequency to minimum.
         Modify_SPI_Reg_bits (0x0117, 10, 0, CONTROL); // write to C_CTL_LPFL_RBB
}
else //high band
{
         low_band = 0; // CONTROL=C_CTL_LPFH_RBB
         CONTROL = 0x7FF; // Set the CONTROL to maximum value. This should bring the output cutt-off frequency to minimum.
         Modify_SPI_Reg_bits (0x0116, 7, 0, CONTROL); // write to C_CTL_LPFH_RBB
}

Set_NCO_Freq (RBB_CalFreq[Band_id]); // Apply a single tone frequency at "CalFreq".

while (1)
{
         ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the value of the amplitude at the ADC input. This value should be lower
than "LowFreqAmp".

         if (ADCOUT >= LowFreqAmp) break; //If it is lower than "LowFreqAmp" repeat cycle
         if (CONTROL == 0) return 0;
         CONTROL--; // Decrease the CONTROL value by one.

         if (low_band) Modify_SPI_Reg_bits (0x0117, 10, 0, CONTROL); // write to C_CTL_LPFL_RBB
                      else Modify_SPI_Reg_bits (0x0116, 7, 0, CONTROL); // write to C_CTL_LPFH_RBB
}

RBB_CBANK[MIMO_ch][Band_id] = CONTROL; // Store CBANK Values
RBB_STATUS[MIMO_ch][Band_id] = 1;
return 1;
}
```

A3.5 TBB calibration

The TBB calibration is divided into two calibrations for low and high bands. Each calibration consists of several smaller algorithms.

A3.6 TBB Low Band Calibration

Calibration steps:

1. Save the current configuration
2. Start with the calibrated value of the R.
3. Approximate (by calculation) the control value of the RBANK for the 11 MHz bandwidth setting.=>Register the value of the RBANK controls (ladder and real pole).
4. Calibrate (by measurement using loopback path 3) the control value of the CBANK(same controls for both the ladder and the real pole) at the 11 MHz bandwidth. => Register the CBANK control value for the low-band section.
5. Calibrate (by measurement using loopback path 4) the mismatch between the pre-emphasis and the real pole stage. => Register the 'high' pre-emphasis parameters. (11 MHz).
6. Calibrate (by measurement using loopback path 5) the control value of the RBANK (ladder only) for the 8.2 MHz bandwidth setting => Register the value of the RBANK controls.

7. Calibrate (by measurement using loopback path 3) the control value of the RBANK (ladder only) for the 5.5 MHz bandwidth setting => Register the value of the RBANK controls.
8. Adjust the value of the real pole controls by -50% (pre-emphasis/real pole RBANK).
9. Calibrate (by measurement using loopback path 4) the mismatch between the pre-emphasis and the real pole stage. => Register the 'low' pre-emphasis parameters. (5.5 MHz).
10. Calibrate (by measurement using loopback path 5) the control value of the RBANK (ladder only) for the 2.74 MHz bandwidth setting => Register the value of the RBANK controls.
11. Calibrate (by measurement using loopback path 5) the control value of the RBANK (ladder only) for the 2.4 MHz bandwidth setting => Register the value of the RBANK controls.
12. Restore the configuration

```
┌──────────────────────────────────────────────┐
│     Calibration_LowBandBand_TBB              │
│     Input Parameters: channel                │
│     Return Parameters: status                │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│     Save current configuration               │
│     Select channel                            │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│     AGC_MODE[1:0](0x040A[13:12])             │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_A_TBB Procedure and get Result │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_B_RBB Procedure for 11 MHz band, │
│  path 3 and get Result                        │
│  Perform Algorithm_C_TBB Procedure and get Result │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_D_RBB Procedure for path 4 and get │
│  Result                                       │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_E_RBB Procedure for 8.2 MHz band, │
│  path 5 and get Result                        │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_E_RBB Procedure for 5.5 MHz band, │
│  path 3 and get Result                        │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Reduce Real Pole RBANK value -50%           │
│  Reduce PreEmphasis Zerp by -50%             │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_D_RBB Procedure for path 4 and get │
│  Result                                       │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_E_RBB Procedure for 2.7 MHz band, │
│  path 5 and get Result                        │
└──────────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────┐
│  Perform Algorithm_E_RBB Procedure for 2.8 MHz band │
│  and get Result                               │
└──────────────────────────────────────────────┘
                      │
                      ▼
              ╭────────────────╮
              │  Return Result. │
              │  END            │
              ╰────────────────╯
```

Figure 36 TBB High Band calibration algorithm

A3.7 TBB High Band Calibration

Calibration steps:

1. Save the current configuration
2. Calibrate (by measurement using loopback path 6) the control value of the RBANK at the18.5 MHz bandwidth. => Register the RBANK control value for the 18.5 MHz.
3. Calibrate (by measurement using loopback path 6) the control value of the RBANK for the 38 MHz bandwidth setting => Register the value of the RBANK controls for 38 MHz.
4. Calibrate (by measurement using loopback path 6) the control value of the RBANK for the 54 MHz bandwidth setting => Register the value of the RBANK controls for 54 MHz.
5. Restore the configuration
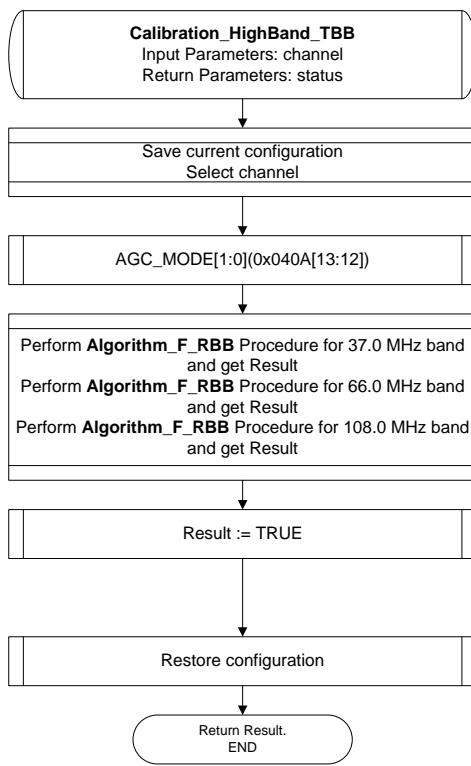
Figure 37 TBB High Band calibration algorithm

The following C code implements the described algorithm:

```
unsigned char Calibration_HighBand_TBB (unsigned char ch)
{
        unsigned char result;

        Save_config_TBB (); //save current configuration

        MIMO_Ctrl (ch);
        Modify_SPI_Reg_bits (0x040A, 13, 12, 1); // AGC Mode = 1 (RSSI mode)

        Set_cal_path_TBB (6); // Set control signals to path 6

        Algorithm_E_TBB (TBB_18_5MHZ);// CalibrateByRes the output cutoff frequency (Algorithm E)
        Algorithm_E_TBB (TBB_38_0MHZ);// CalibrateByRes the output cutoff frequency (Algorithm E)
        Algorithm_E_TBB (TBB_54_0MHZ);// CalibrateByRes the output cutoff frequency (Algorithm E)

        Restore_config_TBB ();      //restore configuration

        return 1;
}
```

A3.8 Nested algorithms

A3.8.1   Algorithm A

Multiply the ratio of the on-chip resistor to the off-chip resistor by the default control value of the RCAL_LPFLAD_TBB for 11 MHz and return the result of the multiplication.

C code for algorithm A:

```
void Algorithm_A_TBB ()
{
            unsigned char RCAL_LPFLAD_TBB;
            float ratio;

            Resistor_calibration (&ratio);
            RCAL_LPFLAD_TBB = (unsigned char)(193 * ratio); // default control value )193 - when 11MHz) Multiply by ratio

            Modify_SPI_Reg_bits (0x0109, 7, 0, RCAL_LPFLAD_TBB);

}
```
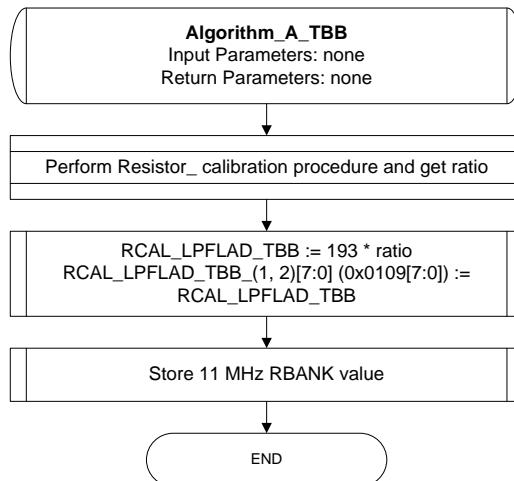

Figure 38 TBB algorithm A

A3.8.2   Algorithm B

This is the same as algorithm B in RBB.

A3.8.3   Algorithm C

Algorithm steps:

1. Apply a single tone frequency at "CalFreq".
2. Set the "CCAL_LPFLAD_TBB" to maximum value.
3. Measure the value of the amplitude at the ADC input. If it is lower than "LowFreqAmp", then jump to line #6 . Otherwise continue.

4. Decrease the control value "CCAL_LPFLAD_TBB" by one step.
5. Jump back to line #3.
6. Store the value of "CCAL_LPFLAD_TBB" as the calibrated CBANK value of TBB.

C code for algorithm C:

```
unsigned char Algorithm_C_TBB (unsigned char Band_id)
{
        unsigned short ADCOUT, LowFreqAmp;
        unsigned char CONTROL;

        Set_NCO_Freq (TBB_CalFreq[Band_id]); // 1 Apply a single tone frequency at "CalFreq".

        CONTROL = 31; // 2 Set the "CCAL_LPFLAD_TBB" to maximum value.
        Modify_SPI_Reg_bits (0x010A, 12, 8, CONTROL);

        while (1)
        {
                ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the value of the amplitude at the ADC input. This value should be lower
than "LowFreqAmp".

                if (ADCOUT >= LowFreqAmp) break; //If amplitude is lower than "LowFreqAmp" repeat cycle
                if (CONTROL == 0) return 0;
                CONTROL--; // Decrease the control value "CCAL_LPFLAD_TBB" by one step.
                Modify_SPI_Reg_bits (0x010A, 12, 8, CONTROL);
        }

        TBB_CBANK[MIMO_ch] = CONTROL; // Store the value of "CCAL_LPFLAD_TBB" as the calibrated CBANK value of TBB.

}
```
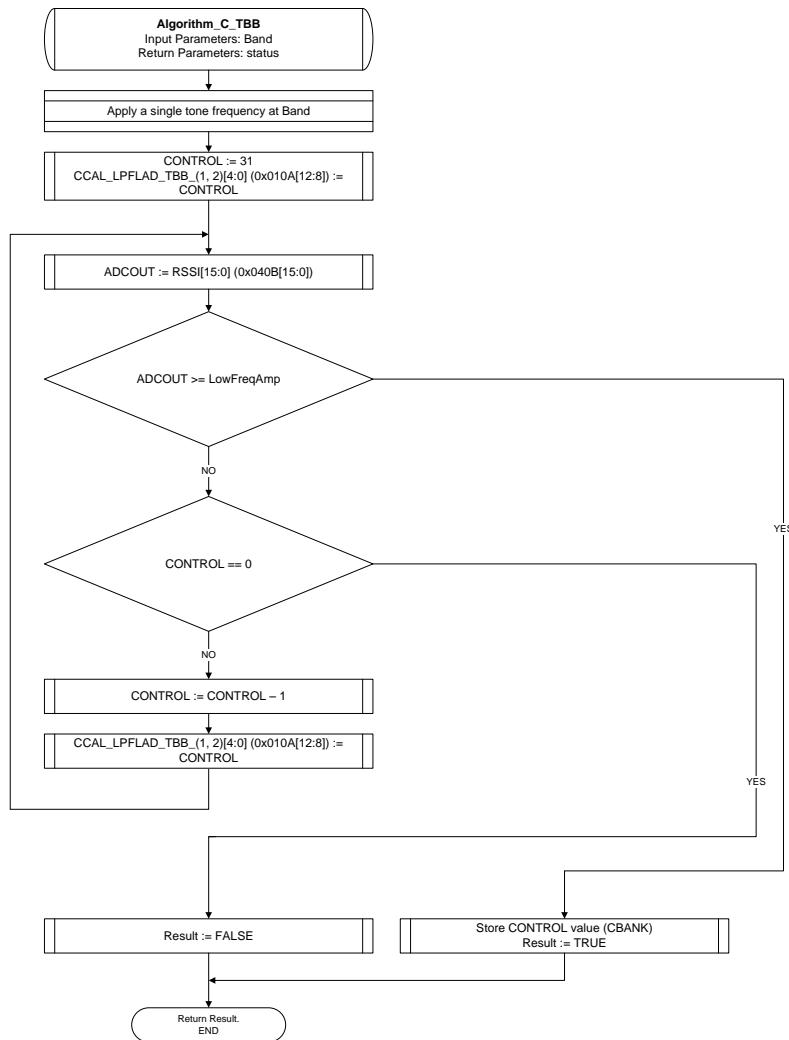

Figure 39 TBB algorithm C

A3.8.4 Algorithm D

Algorithm steps:

1. Apply a single tone at a frequency equal to "CalFreq"
2. Compare the amplitude at the input ADC to "LowFreqAmp". If greater, then the pre-emphasis zero is faster than the real pole. And Vise-Versa. Decrease or increase respectively the zero frequency by one step.
3. If the last step was in the opposite direction of the current step, then you have reached the optimal value of the pre-emphasis parameters. (The last step you increased by one step and this step you decreased, or, the last step you decreased and this step you increased). If not, then go back to step 2.
4. Store the pre-emphasis zero setting.

C code for algorithm D:

```
unsigned char Algorithm_D_TBB (unsigned char Band_id)
{
        unsigned short ADCOUT;
        unsigned char inc, Zero_Freq = 127;

        Set_NCO_Freq (TBB_CalFreq[Band_id]); // 1 Apply a single tone at frequency equal to "CalFreq"

        ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value

        if(ADCOUT > LowFreqAmp) inc = 0; //If greater, then the pre-emphasis zero is faster than the real pole
                else inc = 1;

        while (1)
        {
                ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the output level at the ADC input

                if (inc)
                {
                        if(ADCOUT >= 52428) break;
                }
                else
                {
                        if(ADCOUT <= 52428) break;
                }

                if( (Zero_Freq == 0) || (Zero_Freq == 255))  //gain limit reached
                {
                        return 0;
                        break;
                }

                if(inc) Zero_Freq++;
                        else Zero_Freq--;

        }
}
```

A3.8.5 Algorithm E

1. If("CalFreq") <=11 MHz, then CONTROL=RCAL_LPFLAD_TBB, else, CONTROL=RCAL_LPFH_TBB
2. Set the CONTROL to zero. This should bring the output cut-off frequency to minimum.
3. Apply Algorithm B.
4. Apply a single tone frequency at "CalFreq".
5. Measure the value of the amplitude at the ADC input. This value should be lower than "LowFreqAmp".
6. Increase the CONTROL value by one.
7. Measure the value of the amplitude at the ADC input. If it is lower than "LowFreqAmp", then jump back to line#3. Otherwise continue to step 8.

8. Return the value of CONTROL.
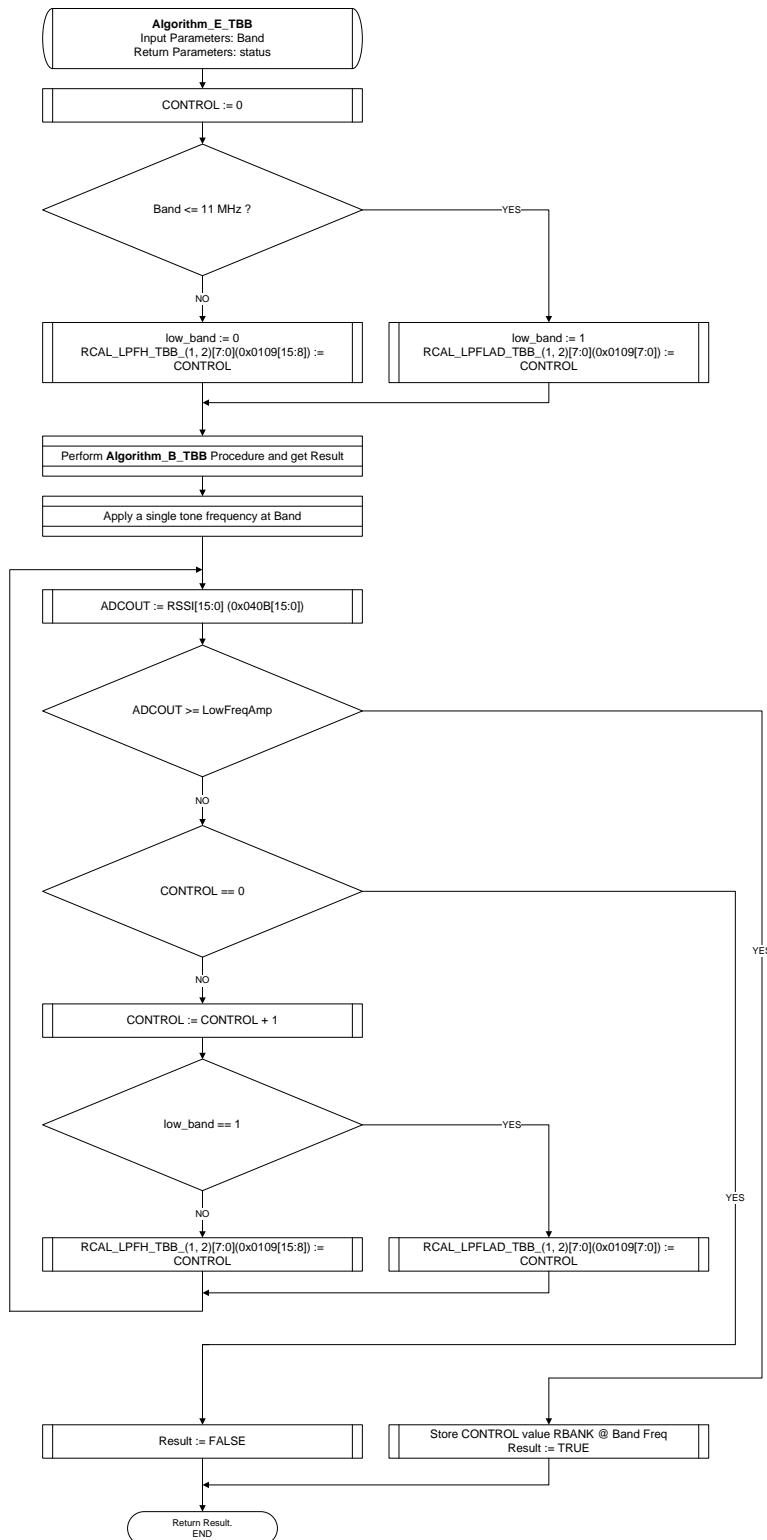
Figure 40 TBB algorithm E

C code for algorithm E:

```
unsigned char Algorithm_E_TBB (unsigned char Band_id)
{
        unsigned short ADCOUT;
        unsigned char low_band, CONTROL;

        CONTROL = 0; // Set the CONTROL to zero. This should bring the output cutt-off frequency to minimum.
```

```
if(Band_id <= TBB_11_0MHZ) //If("CalFreq") <=11MHz, then CONTROL=RCAL_LPFLAD_TBB, else, CONTROL=RCAL_LPFH_TBB
{
            low_band = 1; // CONTROL=RCAL_LPFLAD_TBB
            Modify_SPI_Reg_bits (0x0109, 7, 0, CONTROL); // write to RCAL_LPFLAD_TBB
}
else
{
            low_band = 0; // CONTROL=RCAL_LPFH_TBB
            Modify_SPI_Reg_bits (0x0109, 15, 8, CONTROL); // write to RCAL_LPFH_TBB
}

if (Algorithm_B_TBB (&LowFreqAmp) != 1) return 0; // Calibrate and Record the low frequency output amplitude (Algorithm B)

Set_NCO_Freq (TBB_CalFreq[Band_id]); // Apply a single tone frequency at "CalFreq".

while (1)
{
            ADCOUT = Get_SPI_Reg_bits(0x040B, 15, 0); //RSSI value // Measure the value of the amplitude at the ADC input. This value should be lower
than "LowFreqAmp".

            if (ADCOUT >= LowFreqAmp) break; //If it is lower than "LowFreqAmp" repeat cycle
            if (CONTROL == 0xFF) break;

            CONTROL++; // Increase the CONTROL value by one.

            if (low_band) Modify_SPI_Reg_bits (0x0109, 7, 0, CONTROL); // write to RCAL_LPFLAD_TBB
                        else Modify_SPI_Reg_bits (0x0109, 15, 8, CONTROL); // write to RCAL_LPFH_TBB
}

// 8 Return the value of CONTROL.
TBB_RBANK[MIMO_ch][Band_id] = CONTROL; // Store RBANK Values

return 1;
}
```

NOTICE OF DISCLAMER