Reference manual

The BlueNRG-LP radio controller

Introduction

The BlueNRG-LP BLE controller is a programmable automate, which can act as a master or a slave node compliant up to the Bluetooth Low Energy 5.2 standard. It is an evolution of the previous architecture (BlueNRG-2) keeping the same functionalities and introducing new features that lose even more the bonding between hardware and CPU.

The BLE controller is a coprocessor intended to perform transmission and reception operations without the direct control of the CPU following the instructions included inside some predefined linked lists in RAM. Then, the task of a dedicated link layer firmware is to fill these lists in advance. This allows the controller to start a transaction directly at low power mode exit while the CPU is still booting.

Typically, a processor interrupt service routine runs at the end of every transmitted or received packet in order to prepare/modify the linked lists in RAM or inform the host about changes. When the ISR execution time is a key point, the controller offers the possibility to check if there is enough time to complete the planned transmission/reception and to check the coherency of the RAM lists preventing the reading of not updated data. Furthermore, the controller reads its configuration in three different consecutive phases giving to the CPU the maximum computation time.

The internal sequence of autonomous actions about the transmission or reception is triggered by a timer event that can either wake up the device from deep sleep or not.

A free running counter driven by the internal or external slow clock is always active during sleep mode and provides the absolute time used as reference to schedule the controller activity. It is located in a programmable wake-up block that is also in charge to execute the request from the radio to go into sleep. The CPU is able to program the wake-up block accessing specific registers through the APB interface of the controller.

The BLE controller embeds the following main components:

- A sequencer, which synchronizes the overall link controller. The sequencer controls all the steps from the beginning (one timer trigger) to the last step (interrupt sent to the CPU).
- An AHB arbiter block which allows the communication with an SOC.
- A channel incrementer block which implements in hardware the two-channel incrementing algorithm described in Bluetooth core specification 5.2.
- An AES hardware engine which is able to manage simultaneously three processes:
  - An “On the fly” data packet payload encryption or decryption.
  - An AES 128-bit manual encryption.
  - An LE privacy resolution engine which complies with Bluetooth core specification 5.2.
- A transmit block which takes care of the framing of the advertise or data packet. It supports all coded PHY frame format (with S=2 and S=8) and uncoded PHY frame format (1 Mbps and 2 Mbps). It supports data packet length and advertise packet length up to 255 bytes. The transmit block generates ‘on the fly’ the CRC and the data whitening.
- A receive block which takes care of the data payload de-whitening and the CRC checking for an advertise or data packet. It supports indirectly all coded PHY frame format (with S=2 and S=8) and uncoded PHY frame format (1 Mbps and 2 Mbps). The receive block supports data packet length and advertise packet length up to 255 bytes. It extracts all the header information to ease the sequencer processing.
- An APB main block that manages all the read/write requests coming from the SOC.
- An SN/NESN automatic mechanism, which is automatically disabled when a sequence skip is requested or when no receive buffer in available in RAM.

1 Functional description

Three possible timers (wake-up timer, timer1 and timer2) trigger the start of the controller internal sequence. Wake-up timer and timer1 are based on the absolute machine time. Timer2 is only relative to the end of the previous transmitted or received packet. The wake-up timer is the only timer that is always on during the sleep mode, so it is the only one able to wake up the system when the digital power supply is switched off.

Each time a trigger event is sent to the BLE controller, the sequencer fetches specific tables in RAM to get the required information to know what to configure about the radio and which sequence to start (RX or TX).

There are several types of tables:

- The GlobalStatMach: this table is unique. As the name suggests, it contains valid information for each active link on the device.
- The StatMach: one table for each active link (up to 128 supported by the hardware), called in this context state machine. It contains information such as: the channel, the transmit power and the link to the current action to be performed by the radio, defined by a TxRxPack.
- The TxRxPack: one table for a packet in RX or in TX. There is no predefined number of these tables. They are organized in link lists where each packet points to the next one.
- The DataPack table corresponding to the data buffers pointed by the DataPtr in the TxRxPack. It contains the PDU section of the Bluetooth packet.

After the BLE controller is triggered by one of the three timers, the following six actions define the usual sequence related to a radio transaction.

1. The BLE controller reloads its configuration from retention RAM in order to restore its state (this state might have been modified by the CPU during its interrupt: see action 6 in this list). The link controller reloads its configuration in more separated phases giving more computation time to the CPU.
2. The BLE controller requests the radio access. This action occurs in parallel when action 1 is running.
3. Data transmission or data reception.
4. After the end of all previous actions, the BLE controller writes back its configuration into several tables in RAM and issues an interrupt to the CPU. Depending on internal interrupt enable configuration, the link controller may increase its interrupt (which is connected to the CPU).
5. On interrupt detection, the CPU starts an interrupt service routine (ISR) which checks the controller interrupt status register, reads the configuration saved by the controller in RAM and reads the data received PDU in RAM (in case of valid PDU packet reception). The change from one link state machine to another one is defined by a dedicated function of the ISR, which changes the “CurStMachNum” variable value in the GlobStatMach table.
6. After all the previous activities have ended, the CPU may ask the wake-up block to send a sleep request to the power manager block (PM) to go into sleep mode.

2 Radio resource manager (RRM)

The radio resource manager (RRM) is the block that manages the requests performed by the BLE controller and the CPU to access the radio resources. The requests pass through a semaphore and only one of the two can take control of the radio at a time. The arbitration behaves as follows:

- check the priority value to choose between the BLE controller or the CPU.
- if the same, then the arbiter eliminates the requester that has been served more recently.

The two controllers can request access to the radio resources through a dedicated port:

- Port 0 for the BLE controller
- Port 1 for the CPU (it is a virtual port in this case)

By default, neither the BLE controller nor the CPU has access to the radio. A contributor (BLE or CPU) needs to request a token. The token is requested by software for the CPU. It is done by hardware for the BLE controller each time a timer trig event starts a sequence. Nevertheless, the firmware can release the token granted by the BLE link layer writing inside the CMDREG APB register. Once the requester has the token, its port is granted and it can access the radio resources.

2.1 UDRA

The unified direct register access block allows the software to prepare some commands in a command link list located in the retention RAM. Those commands execute read from and write into the radio registers.

Some interruptions are linked to the UDRA block in the RRM:

- on a command start event
- on a command end event.

The main goal of this block is to allow the BLE controller to reinitialize the radio registers after a low power mode sequence to start an RF communication while the CPU is still being booted.

2.1.1 UDRA RAM command link list

The mapping in RAM for the commands for each port is the following:

- the RadioConfigPtr field of the GlobalStateMach contains the start address of the command start list
- the command start list is a 32-bit element table containing the first command addresses of each command number of each port (more command lists are available for each port)
- each command of each port contains some read and/or write actions on radio registers.

The RadioConfigPtr value is loaded by the RRM-UDRA automatically when the radio controller reset is released. If the software did not initialize this RAM address supposed to point on the command_start_list address before this first automatic load, a “reload pointer” command is available by writing 1 in the UDRA_CTRL0[0] APB register (this bit is auto-cleared immediately).

Note: The RadioConfigPtr pointer value loaded and used by the RRM-UDRA block can be read in the UDRA_RADIO_CFG_PTR APB register.

The port mapping has been defined as follows:

- 2 ports (port0=BLE, port1= VP_CPU)
- port0 supports 3 command lists
- port1 supports 4 command lists.

This leads to a command start list table as presented below:

Table 1. **Command start list details**

Address in RAM	Meaning	Comments
@RadioConfigPtr(value) + 0x00	port0->command0 base address	Command executed by the BLE controller on wake-up timer trigger event if RadioComListEna bit = 1 in on-going StateMach.
@RadioConfigPtr(value) + 0x04	port0->command1 base address	Command executed by the BLE controller on Timer1 trigger event if RadioComListEna bit = 1 in on-going StateMach.
@RadioConfigPtr(value) + 0x08	port0->command2 base address	Command executed by the BLE controller on Timer2 trigger event if RadioComListEna bit = 1 in on-going StateMach.
@RadioConfigPtr(value) + 0x0C	port1->command0 base address	VP_CPU: if the software needs to use an RRM-UDRA command to access the radio register instead of a direct access through APB.
@RadioConfigPtr(value) + 0x10	port1->command1 base address	VP_CPU: if the software needs to use a second RRM-UDRA command to access the radio register instead of a direct access through APB.
@RadioConfigPtr(value) + 0x14	port1->command2 base address	VP_CPU: if the software needs to use a third RRM-UDRA command to access the radio register instead of a direct access through APB.
@RadioConfigPtr(value) + 0x18	port1->command3 base address	VP_CPU: if the software needs to use a fourth RRM-UDRA command to access the radio register instead of a direct access through APB.

2.1.2 UDRA command format in RAM

The write and read command format are described in the following table. Note that only one radio register address is entered for a write or a read. Then, if the number of data to write/read is more than one, the address is incremented automatically by 1.

Table 2. **UDRA command format in RAM**

Byte number	Address in RAM	Byte value	Description
1	command_base_addr	0x--	bit7: 0=write / 1=read bit[6:0] = number of data to write or to read. n = number of data for the example in this table.
2	command_base_addr+1	8-bit address	Address of a Radio register following the 8-bit address mapping.
3	command_base_addr+2	1st data	If write command: write first 8-bit data to be written. If read command: location where the first 8-bit read data are available.
4	command_base_addr+3	2nd data	Optional (depends on number of data to write/read). If write command: write second 8-bit data to be written. If read command: location where the second 8-bit read data are available.
…			
n+2	command_base_addr+(n+1)	nth data	Optional (depends on number of data to write/read). If write command: write n^{th} 8-bit data to be written. If read command: location where the n^{th} 8-bit read data are available.
n+3	command_base_addr+n+2	0x--	Optional: possible to chain other commands. bit7: 0=write / 1=readbit [6:0] = number of data to write or to read.
n+4	command_base_addr+n+3	8-bit address	Address of a radio register following the 8-bit address mapping (see Table 86. Radio Controller registers list)
n+5	command_base_addr+n+4	1st data	If write command: write first 8-bit data to be written. If read command: location where the first 8-bit read data are available.
…			
last	command_base_addr+last-1	0x00 / 0x80	MANDATORY.

Byte number	Address in RAM	Byte value	Description
			The null command (command with null length) must be added at the end of the command list. This is needed by the state machines of the UDRA to be informed they reached the end of the list.

Basic examples:

1) Write AAC0_DIG_ENG=0x12 and AAC1_DIG_ENG=0x34 (grouped registers) through port1.command0:

@port1.command0_addr = 0x02; Write 2 data

@port1.command0_addr+1 = 0x AAC0_DIG_ENG_ADDR;

@port1.command0_addr+2 = 0x12; 1st data to write in AAC0_DIG_ENG

@port1.command0_addr+3 = 0x34; 2nd data to write in AAC1_DIG_ENG

@port1.command0_addr+4 = 0x00; null command

At the end of command execution, the 2 radio registers have been modified with new value.

2) Read of the 4 AFCx_DIG_ENG register chained with a write of 0x54 value in RADIO_FSM_USER through port1.command1:

@port1.command1_addr = 0x84; Read 4 data

@port1.command1_addr+1 = 0xAFC0_DIG_ENG_ADDR

@port1.command1_addr+6 = 0x01; Write 1 data

@port1.command1_addr+7 = 0x RADIO_FSM_USER_ADDR;

@port1.command0_addr+8 = 0x54; 1st (and unique) data to write in RADIO_FSM_USER register

@port1.command0_addr+9 = 0x00; null command

Note: @port1.command1_addr+2 to @port1.command1_addr+5 contents are written by the RRM-UDRA block with the result of read.

At the end of the execution:

- @port1.command1_addr+2 contains the AFC0_DIG_ENG register value
- @port1.command1_addr+3 contains the AFC1_DIG_ENG register value
- @port1.command1_addr+4 contains the AFC2_DIG_ENG register value
- @port1.command1_addr+5 contains the AFC3_DIG_ENG register value
- RADIO_FSM_USER register has been modified with 0x54

2.2 Direct register access

The direct register access block allows the software to access the radio registers directly through an APB access. The radio registers are mainly used to control the analog part of the radio and the radio FSM. The software has to read/write the RF APB registers located inside the RRM APB register list that points directly to the radio registers. The RF APB registers start at RRM address + 0x100. The radio registers are 8-bit only so the APB register bit field [31:8] part is padded with 0. Then, they can be accessed as 32-bit APB registers (address incremented by 4 between each register) through RRM direct access interface. The radio registers can be also accessed exploiting RRM UDRA command list in RAM as 8-bit registers (address incremented by 1 between each register). An internal arbiter manages the case of concurrent accesses on radio registers by both UDRA (executing a command) and direct register access block (on a CPU read/write APB request). The arbitration is based on round-robin priority mechanism. The software must not write any radio registers through direct APB access if they are also modified through commands in RAM (through UDRA block). In this case, there is a risk of multi drivers in parallel and loss of coherency (no way to know which requester wrote the last).

2.2.1 CPU access to radio resources

Although the CPU can request the use of radio resources through the RRM, in most cases read and write accesses to the radio registers can be done directly through APB inside the RRM APB registers. In this case, reading is not intrusive, it is faster and there is no risk even if a radio transaction is on-going. Writing to radio registers is not supposed to be done during a radio transfer. Writing through RRM commands can be safer in order to avoid changes while a radio transaction is on-going. Nevertheless, if this is done very close to a trig event, it is not possible to know which command is executed first between radio and CPU so which setting is used for the coming transaction.

2.3 RRM registers

Table 3. RRM register list

Address offset	Name	RW	Reset	Description
0x00	RRM_ID	R	0x00000001	RRM_ID register
0x04	RRM_CTRL	RW	0x00000003	RRM_CTRL register
0x10	UDRA_CTRL0	RW	0x00000000	UDRA_CTRL0 register
0x14	UDRA_IRQ_ENABLE	RW	0x00000000	UDRA_IRQ_ENABLE register
0x18	UDRA_IRQ_STATUS	RW	0x00000000	UDRA_IRQ_STATUS register
0x1C	UDRA_RADIO_CFG_PTR	R	0x00000000	UDRA_RADIO_CFG_PTR register
0x20	SEMA_IRQ_ENABLE	RW	0x00000000	SEMA_IRQ_ENABLE register
0x24	SEMA_IRQ_STATUS	R	0x00000000	SEMA_IRQ_STATUS register
0x28	BLE_IRQ_ENABLE	RW	0x00000000	BLE_IRQ_ENABLE register
0x2C	BLE_IRQ_STATUS	RW	0x00000000	BLE_IRQ_STATUS register
0x60	VP_CPU_CMD_BUS	RW	0x00000000	VP_CPU_CMD_BUS register
0x64	VP_CPU_SEMA_BUS	RW	0x00000000	VP_CPU_SEMA_BUS register
0x68	VP_CPU_IRQ_ENABLE	RW	0x00000000	VP_CPU_IRQ_ENABLE register
0x6C	VP_CPU_IRQ_STATUS	RW	0x00000000	VP_CPU_IRQ_STATUS register
0x100	AA0_DIG_USR	RW	0x000000D6	AA0_DIG_USR register
0x104	AA1_DIG_USR	RW	0x000000BE	AA1_DIG_USR register
0x108	AA2_DIG_USR	RW	0x00000089	AA2_DIG_USR register
0x10C	AA3_DIG_USR	RW	0x0000008E	AA3_DIG_USR register
0x110	DEM_MOD_DIG_USR	RW	0x00000026	DEM_MOD_DIG_USR register
0x114	RADIO_FSM_USR	RW	0x00000004	RADIO_FSM_USR register
0x118	PHYCTRL_DIG_USR	RW	0x00000000	PHYCTRL_DIG_USR register
0x144	AFC0_DIG_ENG	RW	0x00000066	AFC0_DIG_ENG register
0x148	AFC1_DIG_ENG	RW	0x00000044	AFC1_DIG_ENG register
0x14C	AFC2_DIG_ENG	RW	0x000000FF	AFC2_DIG_ENG register
0x150	AFC3_DIG_ENG	RW	0x0000007F	AFC3_DIG_ENG register
0x154	CR0_DIG_ENG	RW	0x00000044	CR0_DIG_ENG register
0x168	CR0_LR	RW	0x000000DC	CR0_LR register
0x16C	VIT_CONF_DIG_ENG	RW	0x00000000	VIT_CONF_DIG_ENG register
0x184	LR_PD_THR_DIG_ENG	RW	0x00000050	LR_PD_THR_DIG_ENG register
0x188	LR_RSSI_THR_DIG_ENG	RW	0x0000001B	LR_RSSI_THR_DIG_ENG register
0x18C	LR_AAC_THR_DIG_ENG	RW	0x00000038	LR_AAC_THR_DIG_ENG register
0x1DC	DTB0_DIG_ENG	RW	0x00000000	DTB0_DIG_ENG register
0x1F0	DTB5_DIG_ENG	RW	0x00000000	DTB5_DIG_ENG register
0x234	MOD0_DIG_TST	RW	0x00000000	MOD0_DIG_TST register
0x238	MOD1_DIG_TST	RW	0x00000000	MOD1_DIG_TST register
0x23C	MOD2_DIG_TST	RW	0x00000080	MOD2_DIG_TST register

Address offset	Name	RW	Reset	Description
0x240	MOD3_DIG_TST	RW	0x00000098	MOD3_DIG_TST register
0x248	RXADC_ANA_USR	RW	0x0000001B	RXADC_ANA_USR register
0x254	LDO_ANA_ENG	RW	0x00000080	LDO_ANA_ENG register
0x274	CBIAS0_ANA_ENG	RW	0x00000077	CBIAS0_ANA_ENG register
0x278	CBIAS1_ANA_ENG	RW	0x00000007	CBIAS1_ANA_ENG register
0x27C	CBIAS_ANA_TEST	RW	0x00000000	CBIAS_ANA_TEST register
0x280	SYNTHCAL0_DIG_OUT	R	0x00000000	SYNTHCAL0_DIG_OUT register
0x284	SYNTHCAL1_DIG_OUT	R	0x00000001	SYNTHCAL1_DIG_OUT register
0x288	SYNTHCAL2_DIG_OUT	R	0x00000040	SYNTHCAL2_DIG_OUT register
0x28C	SYNTHCAL3_DIG_OUT	R	0x00000000	SYNTHCAL3_DIG_OUT register
0x290	SYNTHCAL4_DIG_OUT	R	0x00000018	SYNTHCAL4_DIG_OUT register
0x294	SYNTHCAL5_DIG_OUT	R	0x00000007	SYNTHCAL5_DIG_OUT register
0x298	FSM_STATUS_DIG_OUT	R	0x00000000	FSM_STATUS_DIG_OUT register
0x29C	IRQ_STATUS_DIG_OUT	R	0x00000000	IRQ_STATUS_DIG_OUT register
0x2A4	RSSI0_DIG_OUT	R	0x00000008	RSSI0_DIG_OUT register
0x2A8	RSSI1_DIG_OUT	R	0x00000008	RSSI1_DIG_OUT register
0x2AC	AGC_DIG_OUT	R	0x00000000	AGC_DIG_OUT register
0x2B0	DEMOD_DIG_OUT	R	0x00000000	DEMOD_DIG_OUT register
0x2B4	AGC0_ANA_TST	RW	0x00000000	AGC0_ANA_TST register
0x2B8	AGC1_ANA_TST	RW	0x00000000	AGC1_ANA_TST register
0x2BC	AGC2_ANA_TST	RW	0x00000000	AGC2_ANA_TST register
0x2C0	AGC0_DIG_ENG	RW	0x0000005E	AGC0_DIG_ENG register
0x2C4	AGC1_DIG_ENG	RW	0x000000CD	AGC1_DIG_ENG register
0x2C8	AGC2_DIG_ENG	RW	0x00000006	AGC2_DIG_ENG register
0x2CC	AGC3_DIG_ENG	RW	0x0000001A	AGC3_DIG_ENG register
0x2D0	AGC4_DIG_ENG	RW	0x00000073	AGC4_DIG_ENG register
0x2D4	AGC5_DIG_ENG	RW	0x0000000F	AGC5_DIG_ENG register
0x2D8	AGC6_DIG_ENG	RW	0x00000000	AGC6_DIG_ENG register
0x2DC	AGC7_DIG_ENG	RW	0x00000000	AGC7_DIG_ENG register
0x2E0	AGC8_DIG_ENG	RW	0x00000000	AGC8_DIG_ENG register
0x2E4	AGC9_DIG_ENG	RW	0x00000090	AGC9_DIG_ENG register
0x2E8	AGC10_DIG_ENG	RW	0x00000000	AGC10_DIG_ENG register
0x2EC	AGC11_DIG_ENG	RW	0x00000000	AGC11_DIG_ENG register
0x2F0	AGC12_DIG_ENG	RW	0x00000000	AGC12_DIG_ENG register
0x2F4	AGC13_DIG_ENG	RW	0x00000000	AGC13_DIG_ENG register
0x2F8	AGC14_DIG_ENG	RW	0x00000000	AGC14_DIG_ENG register
0x2FC	AGC15_DIG_ENG	RW	0x00000000	AGC15_DIG_ENG register
0x300	AGC16_DIG_ENG	RW	0x00000000	AGC16_DIG_ENG register
0x304	AGC17_DIG_ENG	RW	0x00000000	AGC17_DIG_ENG register

Address offset	Name	RW	Reset	Description
0x308	AGC18_DIG_ENG	RW	0x00000000	AGC18_DIG_ENG register
0x30C	AGC19_DIG_ENG	RW	0x00000000	AGC19_DIG_ENG register
0x310	AGC20_DIG_ENG	RW	0x00000080	AGC20_DIG_ENG register
0x324	RXADC_HW_TRIM_OUT	R	0x0000001B	RXADC_HW_TRIM_OUT register
0x328	CBIAS0_HW_TRIM_OUT	R	0x00000078	CBIAS0_HW_TRIM_OUT register
0x32C	CBIAS1_HW_TRIM_OUT	R	0x00000008	CBIAS1_HW_TRIM_OUT register
0x330	AGC_HW_TRIM_OUT	R	0x00000006	AGC_HW_TRIM_OUT register

Table 4. RRM_ID register description

Bit	Field name	Reset	RW	Description
3:0	IDENTIFICATION	0x1	R	RRM Identification register.
31:4	RESERVED31_4	0x0	R	Reserved.

Table 5. RRM_CTRL register description

Bit	Field name	Reset	RW	Description
1:0	PRIORITY	0x3	RW	Defines the priority between direct register or UDRA for radio register access: - 11: Round-robin.
31:2	RESERVED31_2	0x0	R	Reserved.

Table 6. UDRA_CTRL0 register description

Bit	Field name	Reset	RW	Description
0	RELOAD_RDCFGPTR	0x0	RW	Reload the radio configuration pointer from RAM. This bit is auto-cleared by hardware.
31:1	RESERVED31_1	0x0	R	Reserved.

Table 7. UDRA_IRQ_ENABLE register description

Bit	Field name	Reset	RW	Description
0	RADIO_CFG_PTR_RELOADED	0x0	RW	UDRA interrupt enable (reload radio config pointer).
1	CMD_START	0x0	RW	UDRA interrupt enable (command start).
2	CMD_END	0x0	RW	UDRA interrupt enable (command end).
3	CMD_NUMBER_ERROR	0x0	RW	UDRA interrupt enable (error in the number of command).
31:4	RESERVED31_4	0x0	R	Reserved.

Table 8. **UDRA_IRQ_STATUS register description**

Bit	Field name	Reset	RW	Description
0	RADIO_CFG_PTR_RELOADED	0x0	RW	UDRA interrupt enable (reload radio config pointer). Write '1' to clear IRQ status bit.
1	CMD_START	0x0	RW	UDRA interrupt enable (command start). Write '1' to clear IRQ status bit.
2	CMD_END	0x0	RW	UDRA interrupt enable (command end). Write '1' to clear IRQ status bit.
3	CMD_NUMBER_ERROR	0x0	RW	UDRA interrupt enable (error in the number of command). Write '1' to clear IRQ status bit.
31:4	RESERVED31_4	0x0	R	Reserved.

Table 9. **UDRA_RADIO_CFG_PTR register description**

Bit	Field name	Reset	RW	Description
31:0	RADIO_CONFIG_ADDRESS	0x0	R	UDRA radio configuration address. This field contains the value contained by RadioConfigPtr bit field in GlobalStatMach RAM table when the BLE controller exits the reset state. This field is updated after a reload configuration pointer command.

Table 10. **SEMA_IRQ_ENABLE register description**

Bit	Field name	Reset	RW	Description
0	LOCK	0x0	RW	Semaphore locked (= one port granted) interrupt enable.
1	UNLOCK	0x0	RW	Semaphore unlocked (= no port selected) interrupt enable.
31:2	RESERVED31_2	0x0	R	Reserved.

Table 11. **SEMA_IRQ_STATUS register description**

Bit	Field name	Reset	RW	Description
0	LOCK	0x0	R	On read, returns the semaphore locked interrupt status. Write '1' to clear this IRQ status bit.
1	UNLOCK	0x0	R	On read, returns the semaphore unlocked interrupt status. Write '1' to clear this IRQ status bit.
31:2	RESERVED31_2	0x0	R	Reserved.

Table 12. **BLE_IRQ_ENABLE register description**

Bit	Field name	Reset	RW	Description
0	PORT_GRANT	0x0	RW	BLE port grant interrupt enable.
1	PORT_RELEASE	0x0	RW	BLE port release interrupt enable.
2	PORT_PREEMPT	0x0	RW	BLE port preempt interrupt enable.
3	PORT_CMD_START	0x0	RW	BLE port command start interrupt enable.
4	PORT_CMD_END	0x0	RW	BLE port command end interrupt enable.
31:5	RESERVED31_5	0x0	R	Reserved

Table 13. BLE_IRQ_STATUS register description

Bit	Field name	Reset	RW	Description
0	PORT_GRANT	0x0	RW	BLE hardware port granted interrupt status. - 0: the BLE port request to semaphore is not granted. - 1: the BLE controller request to take the semaphore is granted: the RF registers access and the radio TX and the radio RX data path are selected for that controller. The port stays granted as long as it requests the token and the semaphore is not preempted by another port. Write '1' to clear this IRQ status bit.
1	PORT_RELEASE	0x0	RW	BLE hardware port released interrupt status. When read: - 0: the BLE controller has not been released. - 1: the BLE controller has been released by the semaphore. Write '1' to clear this IRQ status bit.
2	PORT_PREEMPT	0x0	RW	BLE hardware port preemption (at semaphore level) interrupt status. When read: - 0: the BLE controller has not been preempted by another controller. - 1: the BLE controller has been preempted and semaphore token was taken by another port. Write '1' to clear this IRQ status bit.
3	CMD_START	0x0	RW	BLE hardware port command start interrupt status. When read: - 0: the BLE port command requested by the BLE controller is not started. - 1: the BLE port command requested by the BLE controller is started. Write '1' to clear this IRQ status bit.
4	CMD_END	0x0	RW	BLE hardware port command end interrupt status. When read: - 0: the BLE port command requested by the BLE controller is not completed. - 1: the BLE port command requested by the BLE controller is completed. Write '1' to clear this IRQ status bit.
31:5	RESERVED31_5	0x0	R	Reserved

Note: The BLE controller receives the previous information directly by hardware wires and manages the sequence through them. The interrupt mechanism is there in case the CPU needs to monitor the activity between the BLE controller and the RRM block.

Table 14. VP_CPU_CMD_BUS register description

Bit	Field name	Reset	RW	Description
2:0	COMMAND	0x0	RW	Command number.
3	COMMAND_REQ	0x0	RW	CPU Virtual port command request - 0: the RRM command request is released. - 1: request a command to the RRM-UDRA block. This bit is cleared by HW once the command is ended.
31:4	RESERVED31_4	0x0	R	Reserved.

Table 15. VP_CPU_SEMA_BUS register description

Bit	Field name	Reset	RW	Description
2:0	TAKE_PRIO	0x0	RW	Semaphore priority value (between 0 and 7) of the take request. The higher the value, the higher priority is the request.
3	TAKE_REQ	0x0	RW	Semaphore token request - 0: the CPU virtual port releases the semaphore or does not request to take the RRM semaphore. - 1: the CPU virtual port requests to take or to keep the RRM semaphore.
4	TAKE_PREEMPT	0x0	RW	Semaphore token preemption request by the CPU virtual port - 0: Semaphore take request is not applied with pre-emption. This is the usual use case to request the semaphore. - 1: Semaphore take request is applied with preemption. TAKE_PREEMPT=1'b1 should only be used exceptionally when the peripheral cannot wait anymore to get the semaphore due to timing constraint of a radio protocol.
31:5	RESERVED31_5	0x0	R	Reserved

Table 16. VP_CPU_IRQ_ENABLE register description

Bit	Field name	Reset	RW	Description
0	PORT_GRANT	0x0	RW	CPU virtual port grant interrupt enable.
1	PORT_RELEASE	0x0	RW	CPU virtual port release interrupt enable.
2	PORT_PREEMPT	0x0	RW	CPU virtual port preempt interrupt enable.
3	PORT_CMD_START	0x0	RW	CPU virtual port command start interrupt enable.
4	PORT_CMD_END	0x0	RW	CPU virtual port command end interrupt enable.

Table 17. **VP_CPU_IRQ_STATUS register description**

Bit	Field name	Reset	RW	Description
0	PORT_GRANT	0x0	RW	CPU virtual port granted interrupt status. - 0: the CPU virtual port token request is not granted. - 1: the CPU virtual port token request is granted by the semaphore: Write '1' to clear this IRQ status bit.
1	PORT_RELEASE	0x0	RW	CPU virtual port released interrupt status. - 0: the CPU virtual port has not been released (due to TAKE_REQ=1'b1) - 1: the CPU virtual port has been released by the semaphore due to TAKE_REQ=1'b0 (requested by CPU virtual port). Write '1' to clear this IRQ status bit.
2	PORT_PREEMPT	0x0	RW	CPU virtual port preemption (at semaphore level) interrupt status. When read: - 0: the CPU virtual port has not been preempted by another UDRA port. - 1: the CPU virtual port has been preempted by another UDRA port. Write '1' to clear this IRQ status bit.
3	CMD_START	0x0	RW	CPU virtual port command start interrupt status. When read: - 0: the command requested by the CPU virtual port (port1) is not started. - 1: the command requested by the CPU virtual port (port1) is started Write '1' to clear this IRQ status bit.
4	CMD_END	0x0	RW	CPU virtual port command end interrupt status. When read: - 0: the command requested by the CPU virtual port (port1) is not completed. - 1: the command requested by the CPU virtual port (port1) is completed. Write '1' to clear this IRQ status bit.

2.3.1 Radio registers (RRM address + 0x100)

They can be accessed through two different mappings:

- as 32-bit APB register (address incremented by 4 between each register) through RRM direct access interface as 8-bit register (address incremented by 1 between each register) through RRM UDRA command list in RAM.

Table 18. **AA0_DIG_USR register description**

Bit	Field name	Reset	RW	Description
7:0	AA_7_0	0xD6	RW	Least significant byte of the BTLE Access Address code. This register is (over)written by the sequencer during 2nd INIT step with the StatMach.accaddr[7:0] bit field.
31:8	RESERVED31_8	0x0	R	Reserved

Table 19. AA1_DIG_USR register description

Bit	Field name	Reset	RW	Description
7:0	AA_15_8	0xBE	RW	Next byte of the BTLE access address code. This register is (over)written by the sequencer during 2nd INIT step with the StatMach.accaddr[15:8] bit field.
31:8	RESERVED31_8	0x0	R	Reserved

Table 20. AA2_DIG_USR register description

Bit	Field name	Reset	RW	Description
7:0	AA_23_16	0x89	RW	Next byte of the BTLE access address code This register is (over)written by the sequencer during 2nd INIT step with the StatMach.accaddr[23:16] bit field.
31:8	RESERVED31_8	0x0	R	Reserved

Table 21. AA3_DIG_USR register description

Bit	Field name	Reset	RW	Description
7:0	AA_31_24	0x89	RW	Next byte of the BTLE access address code. This register is (over)written by the sequencer during 2nd INIT step with the StatMach.accaddr[31:24] bit field.
31:8	RESERVED31_8	0x0	R	Reserved

Table 22. DEM_MOD_DIG_USR register description

Bit	Field name	Reset	RW	Description
0	SPARE	0x0	RW	Spare
7:1	CHANNEL_NUM	0x13	RW	Index for internal lock-up table in which the synthesizer setup is contained. Default value is the BLE RF channel 19 -> 2440 MHz. For Bluetooth protocol: this bit field is (over)written by the BLE sequencer during the 1st INIT. The value copied here is the output of the channel Incr and hoping hardware block. Example: value to program to select the channel 19: CHANNEL_NUM = 19 = 0x13. Then, 2402 + (channel number * 2) = 2440 MHz for BLE channel 19 Note: This bit field is used by the SYNTH_IF hardware block to generate the physical frequency on the antenna.
31:8	RESERVED31_8	0x0	R	Reserved

Table 23. RADIO_FSM_USR register description

Bit	Field name	Reset	RW	Description
0	TXMODE	0x0	RW	Tx mode bit. For Bluetooth protocol, this bit is (over)written by the BLE sequencer with the StatMach.TxMode bit during the 1st INIT step. Note: This bit is not used by the hardware.
1	EN_CALIB_CBP	0x0	RW	CBP calibration enable bit. For Bluetooth protocol, this bit is (over)written by the BLE sequencer with the TxRxPack.CalReq bit during the 1st INIT step. Note: This bit is used by the radio FSM as EN_CALIB_SYNTH information.
2	EN_CALIB_SYNTH	0x1	RW	SYNTH calibration enable bit. For Bluetooth protocol, this bit is (over)written by the BLE sequencer with the TxRxPack.CalReq bit during the 1st INIT step. Note: This bit is used by the Radio FSM as EN_CALIB_SYNTH information.
7:3	PA_POWER	0x0	RW	PA power coefficient. For Bluetooth protocol, this bit is (over)written by the BLE sequencer with the StatMach.PAPower bit field during the 1st INIT step. Note: This bit is used by the PA_RAMP hardware block.
31:8	RESERVED31_8	0x0	R	Reserved

Table 24. PHYCTRL_DIG_USR register description

Bit	Field name	Reset	RW	Description
2:0	RXTXPHY	0x0	RW	RXTXPHY selection. For Bluetooth protocol, this bit field is (over)written by the BLE sequencer during the 1st INIT using the StatMach.RxPhy[2:0] or StatMach.TxPhy[2:0], depending if the transfer is a reception or a transmission. - 000: uncoded PHY 1 Mb/s - 001: uncoded PHY 2 Mb/s - 100: coded PHY S=8 1 Mb/s - 110: coded PHY S=2 1 Mb/s Note: This bit field is used by the hardware to inform the digital and analog blocks needing this PHY information.
3:7	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 25. AFC0_DIG_ENG register description

Bit	Field name	Reset	RW	Description
3:0	AFC_GAIN_AFTER	0x6	RW	Set the gain of the AFC loop before AA detection to the value2^(-AFC_GAIN_AFTER).
7:4	AFC_GAIN_BEFORE	0x6	RW	Set the gain of the AFC loop before AA detection to the value2^(-AFC_GAIN_BEFORE).
31:8	RESERVED31_8	0x0	R	Reserved.

Table 26. AFC1_DIG_ENG register description

Bit	Field name	Reset	RW	Description
3:0	AFC_DELAY_AFTER	0x4	RW	Set the gain of the AFC loop before AA detection to the value AFC_DELAY_AFTER/256.
7:4	AFC_DELAY_BEFORE	0x4	RW	Set the decay factor of the AFC loop before AA detection to the value AFC_DELAY_BEFORE/ 256.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 27. AFC2_DIG_ENG register description

Bit	Field name	Reset	RW	Description
6:0	AFC_FREQ_LIMIT	0x7F	RW	Max. (absolute value) of frequency correction.
7	AFC_ENABLE	0x1	RW	Enable/disable the AFC loop - 0: disabled - 1: enabled
31:8	RESERVED31_8	0x0	R	Reserved.

Table 28. AFC3_DIG_ENG register description

Bit	Field name	Reset	RW	Description
7:0	AFC_MINMAX_LIMIT	0x7F	RW	Max. difference allowed on the min./max. peak detectors. Values above this limit are interpreted as noise and the current min./max. trackers are reset.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 29. CR0_DIG_ENG register description

Bit	Field name	Reset	RW	Description
3:0	CR_GAIN_AFTER	0x4	RW	Set the gain of the clock recovery loop before AA detection to the value 2^(-CR_GAIN_AFTER).
7:4	CR_GAIN_BEFORE	0x4	RW	Set the gain of the clock recovery loop before AA detection to the value 2^(-CR_GAIN_BEFORE).
31:8	RESERVED31_8	0x0	R	Reserved.

Table 30. CR0_LR register description

Bit	Field name	Reset	RW	Description
3:0	CR_LR_GAIN_AFTER	0x6	RW	Set the gain of the clock recovery loop before AA detection to the value 2^(-CR_GAIN_BEFORE).
7:4	CR_LR_GAIN_BEFORE	0x6	RW	Set the gain of the clock recovery loop before AA detection to the value 2^(-CR_GAIN_BEFORE).
31:8	RESERVED31_8	0x0	R	Reserved.

Table 31. VIT_CONF_DIG_ENG register description

Bit	Field name	Reset	RW	Description
7:0	VIT_CONF	0x0	RW	Viterbi control register - VIT_CONF[0] = enable the Viterbi - VIT_CONF[1] = PD_DETECT_MODE: Preamble detection mode selection (0 = Peak repetition, 1 = RSSI)
31:8	RESERVED31_8	0x0	R	Reserved

Table 32. LR_PD_THR_DIG_ENG register description

Bit	Field name	Reset	RW	Description
7:0	LR_PD_THR	0x50	RW	Preamble detect threshold value
31:8	RESERVED31_8	0x0	R	Reserved.

Table 33. LR_RSSI_THR_DIG_ENG register description

Bit	Field name	Reset	RW	Description
7:0	LR_RSSI_THR	0x1B	RW	RSSI or peak threshold value.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 34. LR_AAC_THR_DIG_ENG register description

Bit	Field name	Reset	RW	Description
7:0	LR_RSSI_THR	0x1B	RW	Address coded correlation threshold.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 35. DTB0_DIG_ENG register description

Bit	Field name	Reset	RW	Description
0	DTB_EN	0x0	RW	Enable DTB.
4:1	DTB_CFG	0x0	RW	DTB configuration.
7:5	SPARE	0x0	RW	Spare.
31:8	RESERVED31_5	0x0	R	Reserved.

Table 36. DTB5_DIG_ENG register description

Bit	Field name	Reset	RW	Description
0	RXTX_START_SEL	0x0	RW	Selection 1 enable.
1	TX_ACTIVE	0x0	RW	Force TX_ACTIVE signal.
2	RX_ACTIVE	0x0	RW	Force RX_ACTIVE signal.
3	INITIALIZE	0x0	RW	Force INITIALIZE signal.
4	PORT_SELECTED_EN	0x0	RW	Enable port selection.
5	PORT_SELECTED_0	0x0	RW	Force port_selected[0] signal.
7:6	SPARE	0x0	RW	Spare.
31:7	RESERVED31_6	0x0	R	Reserved.

Table 37. MOD0_DIG_TST register description

Bit	Field name	Reset	RW	Description
0	MOD_DIG_TEST_SEL	0x0	RW	Selection - 0: forced by modulator (normal mode) - 1: not forced by modulator but by MODx_TST registers values (test mode)
2:1	SPARE	0x0	RW	Spare
3	PMU_NO_MODULTATION	0x0	RW	Bypass modulation - 0: no bypass - 1: bypass
7:4	KFORCE_3_0	0x0	RW	Fraction part.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 38. MOD1_DIG_TST register description

Bit	Field name	Reset	RW	Description
7:0	KFORCE_11_4	0x0	RW	Fraction part.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 39. MOD2_DIG_TST register description

Bit	Field name	Reset	RW	Description
7:0	KFORCE_19_12	0x80	RW	Fraction part.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 40. MOD3_DIG_TST register description

Bit	Field name	Reset	RW	Description
2:0	AFORCE	0x0	RW	Integer part.
7:3	MFORCE	0x13	RW	Integer part.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 41. RXADC_ANA_USR register description

Bit	Field name	Reset	RW	Description
2:0	RFD_RXADC_DELAYTRIM_I	0x3	RW	ADC loop delay control bits for I channel.
5:3	RFD_RXADC_DELAYTRIM_Q	0x3	RW	ADC loop delay control bits for Q channel.
6	RXADC_DELAYTRIM_I_TST_SEL	0x0	RW	When set, RFD_RXADC_DELAYTRIM_I[2:0] bit field is used instead of the HW trimming.
7	RXADC_DELAYTRIM_Q_TST_SEL	0x0	RW	When set, RFD_RXADC_DELAYTRIM_Q[2:0] bit field is used instead of the HW trimming.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 42. **LDO_ANA_ENG register description**

Bit	Field name	Reset	RW	Description
0	SPARE	0x0	RW	Spare.
1	RFD_LDO_TRANSFO_BYPASS	0x0	RW	VDD level Bypass mode - 0: Bypass mode disabled - 1: LDO in bypass mode.
2	RFD_LDO_RXADC_BYPASS	0x0	RW	VDD level Bypass mode - 0: Bypass mode disable - 1: LDO in Bypass mode.
3	RFD_LDO_RX_TX_BYPASS	0x0	RW	VDD level Bypass mode - 0: Bypass mode disable - 1: LDO in Bypass mode.
7:4	SPARE	0x8	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 43. **CBIAS0_ANA_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	RFD_CBIAS_IBIAS_TRIM	0x7	RW	Ibias current trimming.
7:4	RFD_CBIAS_IPTAT_TRIM	0x7	RW	Ibias current trimming.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 44. **CBIAS1_ANA_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	RFD_CBIAS_VBG_TRIM	0x7	RW	Ibias current trimming.
4	RFD_CBIAS_ENA_ATB_CURR	0x0	RW	Enable CB for ATB - 0: disable - 1: enable.
5	CBIAS_CURR2_PREBOOST	0x0	RW	Select the moment to activate the current 2 section: - 0: RFD_CBIAS_SEL_CURR_2 active from ENA_LDO state of the Radio FSM - 1: RFD_CBIAS_SEL_CURR_2 active from VBG_BOOST state of the radio FSM.
6	CBIAS_VBG_TRIM_TST_SEL	0x0	RW	Select the VBG trimming value source: - 0: trimming applied on the analog block are the HW loaded ones - 1: trimming applied on the analog block are provided by the RFD_CBIAS_VBG_TRIM bit fields (SW values).
7	CBIAS0_TRIM_TST_SEL	0x0	RW	Select the CBIAS IPTAT and IBIAS trimming values source: - 0: trimming applied on the analog block are the HW loaded ones - 1: trimming applied on the analog block are provided by the CBIAS0_ANA_ENG bit fields (SW values).
31:8	RESERVED31_8	0x0	R	Reserved.

Table 45. CBIAS_ANA_TEST register description

Bit	Field name	Reset	RW	Description
0	CBIAS_ANA_TST_SEL	0x0	RW	Selection - 0: default value is 0 - 1: forced by this register.
1	RESERVED	0x0	RW	Reserved.
2	RFD_CBIAS_ENA_CORE	0x0	RW	Enable core central bias - 0: disable - 1: enable
3	RFD_CBIAS_ENA_RX_CURR	0x0	RW	Enable RX and TX central bias - 0: disable - 1: enable
4	RFD_CBIAS_ENA_TX_CURR	0x0	RW	Not connected inside the design. This bit has no effect.
5	RFD_CBIAS_ENA_NF_OFF	0x0	RW	Spare.
6	RFD_CBIAS_ENA_VBG_BOOST	0x0	RW	VBG boost enable - 0: disable - 1: enable
7	RFD_CBIAS_ENA_VBG	0x0	RW	VBG enable - 0: disable - 1: enable
31:8	RESERVED31_8	0x0	R	Reserved.

Table 46. SYNTHCAL0_DIG_OUT register description

Bit	Field name	Reset	RW	Description
6:0	VCO_CALAMP_OUT_6_0	0x0	R	VCO CALAMP value.
7	RESERVED7	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 47. SYNTHCAL1_DIG_OUT register description

Bit	Field name	Reset	RW	Description
3:0	VCO_CALAMP_OUT_10_7	0x1	R	VCO CALAMP value.
7:4	SPARE	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 48. SYNTHCAL2_DIG_OUT register description

Bit	Field name	Reset	RW	Description
6:0	VCO_CALFREQ_OUT	0x40	R	VCO CALFREQ value.
7	RESERVED7	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 49. SYNTHCAL3_DIG_OUT register description

Bit	Field name	Reset	RW	Description
3:0	SYNTHCAL_DEBUG_BUS	0x0	R	Calibration debug bus.
7:4	RESERVED7_4	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 50. SYNTHCAL4_DIG_OUT register description

Bit	Field name	Reset	RW	Description
5:0	MOD_REF_DAC_WORD_OUT	0x18	R	Calibration word.
7:6	SPARE	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 51. . SYNTHCAL5_DIG_OUT register description

Bit	Field name	Reset	RW	Description
3:0	CBP_CALIB_WORD	0x7	R	CBP calibration word.
7:4	RESERVED7_4	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 52. FSM_STATUS_DIG_OUT register description

Bit	Field name	Reset	RW	Description
4:0	STATUS	0x0	R	STATUS: RF FSM state: - 00000: IDLE - 00001: ACTIVE1 - 00010: VBG_BOOST - 00011: ENA_CURR - 00100: ACTIVE2 - 00101 to 01111: Not used - 10000: ENA_LDO - 10001: SYNTH_SETUP - 10010: CALIB10 - 10011: CALIB01 - 10100: CALIB11 - 10101: LOCKRXTX - 10110: Not used - 10111: Not used - 11000: EN_RX - 11001: EN_PA - 11010: RX - 11011: RX_802_RESET - 11100: TX - 11101: Not used - 11110: PA_DWN_ANA - 11111: Not used
6:5	RESERVED6_5	0x0	R	Reserved.
7	SYNTH_CAL_ERROR	0x0	R	PLL calibration error.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 53. IRQ_STATUS_DIG_OUT register description

Bit	Field name	Reset	RW	Description
7:0	RESERVED7_0	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 54. RSSI0_DIG_OUT register description

Bit	Field name	Reset	RW	Description
7:0	RSSI_MEAS_OUT_7_0	0x8	R	Measure of the received signal strength.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 55. **RSSI1_DIG_OUT register description**

Bit	Field name	Reset	RW	Description
7:0	RSSI_MEAS_OUT_15_8	0x8	R	Measure of the received signal strength.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 56. **AGC_DIG_OUT register description**

Bit	Field name	Reset	RW	Description
3:0	AGC_ATT_OUT	0x0	R	AGC attenuation value.
7:4	RESERVED7_4	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 57. **DEMOD_DIG_OUT register description**

Bit	Field name	Reset	RW	Description
1:0	CI_FIELD	0x0	R	CI field
2	AAC_FOUND	0x0	R	aac_found
3	PD_FOUND	0x0	R	pd_found
4	RX_END	0x0	R	rx_end
7:5	RESERVED7_5	0x0	R	Reserved.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 58. **AGC0_ANA_TST register**

Bit	Field name	Reset	W	Description
0	AGC0_ANA_TST_SEL	0x0	RW	Selection - 0: default value is 0 (normal mode), - 1: forced by register (test mode).
3:1	AGC_ANT	0x0	RW	AGC on antenna.
4	AGC_LNA	0x0	RW	AGC on LNA.
7:5	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 59. **AGC1_ANA_TST register description**

Bit	Field name	Reset	RW	Description
0	AGC1_ANA_TST_SEL	0x0	RW	Selection - 0: default value is 0 (normal mode), - 1: forced by this register (test mode).
5:1	AGC_IFATT	0x0	RW	AGC on IF ATT.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 60. **AGC2_ANA_TST register description**

Bit	Field name	Reset	RW	Description
0	AGC2_ANA_TST_SEL	0x0	RW	Selection - 0: default value is 0 (normal mode): the AGC antenna trimming value comes from the SoC. - 1: forced by this register (test mode): the AGC antenna trim value comes from the AGC2_ANA_RST[3:1] bit field value.
3:1	AGC_ANTENNAE_USR_TRIM	0x0	RW	AGC trimming.
7:4	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 61. **AGC0_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	AGC_THR_HIGH	0x1E	RW	High AGC threshold.
6	AGC_ENABLE	0x1	RW	Enable AGC.
7	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 62. **AGC1_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	AGC_THR_LOW_6	0xD	RW	Low threshold for 6dB steps.
6	AGC_AUTOLOCK	0x1	RW	AGC locks when level is steady between high threshold and lock threshold.
7	AGC_LOCK_SYNC	0x1	RW	AGC locks when AA is detected.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 63. **AGC2_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	AGC_THR_LOW_12	0x6	RW	Low AGC threshold for 12 dB steps.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 64. **AGC3_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	AUTOLOCK_THR	0x1A	RW	Threshold for autolock.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 65. **AGC4_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	AGC_HOLD_TIME_FAST	0x3	RW	AGC hold time for fast transitions.
7:4	AGC_HOLD_TIME_SLOW	0x7	RW	AGC hold time for slow transitions.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 66. **AGC5_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	T_MEAS	0xF	RW	Measurement time.
7:4	T_INT	0x0	RW	Duration for AGC initial wait phase.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 67. **AGC6_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
6:0	HOLD_TIME_SEL_10_4	0x0	RW	Hold time selection bit.
7	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 68. **AGC7_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
6:0	TH_LOW_SEL_10_4	0x0	RW	Low threshold selection bit.
7	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 69. **AGC8_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	HOLD_TIME_SEL_3_0	0x0	RW	Hold time selection bit.
7:4	TH_LOW_SEL_3_0	0x0	RW	Low threshold selection bit.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 70. **AGC9_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
3:0	START_SEQ	0x0	RW	Initial AGC value.
7:4	MAX_SEQ	0x9	RW	Maximum value for the AGC value.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 71. AGC10_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_0	0x0	RW	Mapping for AGC step 0.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 72. AGC11_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_1	0x0	RW	Mapping for AGC step 1.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 73. AGC12_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_2	0x0	RW	Mapping for AGC step 2.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 74. AGC13_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_3	0x0	RW	Mapping for AGC step 3.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 75. AGC14_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_4	0x0	RW	Mapping for AGC step 4.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 76. AGC15_DIG_ENG register description

Bit	Field name	Reset	RW	Description
5:0	ATT_5	0x0	RW	Mapping for AGC step 5.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 77. **AGC16_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	ATT_6	0x0	RW	Mapping for AGC step 6.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 78. **AGC17_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	ATT_7	0x0	RW	Mapping for AGC step 7.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 79. **AGC18_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	ATT_8	0x0	RW	Mapping for AGC step 8.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 80. **AGC19_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
5:0	ATT_9	0x0	RW	Mapping for AGC step 9.
7:6	SPARE	0x0	RW	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 81. **AGC20_DIG_ENG register description**

Bit	Field name	Reset	RW	Description
7:0	I_GAIN_COMP	0x80	RW	Gain compensation for I branch.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 82. **RXADC_HW_TRIM_OUT register description**

Bit	Field name	Reset	RW	Description
2:0	HW_RXADC_DELAYTRIM_I	0x3	R	Control bits of the RX ADC loop delay for I channel (provided by the HW trimming, automatically loaded on POR).
5:3	HW_RXADC_DELAYTRIM_Q	0x3	R	Control bits of the RX ADC loop delay for Q channel (provided by the HW trimming, automatically loaded on POR).
7:6	SPARE	0x0	R	Spare.
31:8	RESERVED31_8	0x0	R	Reserved.

Table 83. CBIAS0_HW_TRIM_OUT register description

Bit	Field name	Reset	RW	Description
3:0	HW_CBIAS_IBIAS_TRIM	0x8	R	IBIAS current (provided by the HW trimming, automatically loaded on POR).
7:4	HW_CBIAS_IPTAT_TRIM	0xE	R	IPTAT current (provided by the HW trimming, automatically loaded on POR).
31:8	RESERVED31_8	0x0	R	Reserved.

Table 84. CBIAS1_HW_TRIM_OUT register description

Bit	Field name	Reset	RW	Description
3:0	HW_CBIAS_VBG_TRIM	0x8	R	VBG current (provided by the HW trimming, automatically loaded on POR).
31:4	RESERVED31_8	0x0	R	Reserved.

Table 85. AGC_HW_TRIM_OUT register description

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	R	Reserved.
3:1	HW_AGC_ANTENNAE_TRIM	0x6	R	AGC trim value (provided by the HW trimming, automatically loaded on POR). Note: This value depends on the RF BOM on the board. Value provided by engineering is based on a dedicated BOM and must be overloaded by SW if the user selects/defines another BOM.
31:4	RESERVED31_4	0x0	R	Reserved.

3 Radio FSM

The radio FSM manages the startup and stop sequences of the analog part of the radio depending on requesting RF transfer.

3.1 Radio FSM sequences

This paragraph lists the main steps in radio FSM sequence.

- The radio FSM stays in IDLE as long as the IP_BLE does not request the RRM token to indicate the radio is about to be used.
- Once the token is requested, the radio FSM switches to ACTIVE1.
- When the device switches on the accurate fast clock (external XO) AND if the RRM semaphore granted one port (whatever the port), the radio FSM goes to ACTIVE2 (through a few intermediate steps to start bandgap and central bias current).
- Once an RX or TX request is received, the radio FSM switches to the RX or TX through intermediate steps to setup properly the analog.
- The radio FSM goes back to ACTIVE2 as soon as RX or TX request is cleared and back to ACTIVE1 if the accurate clock is replaced by the dirty one or if no more ports request a token to the RRM semaphore.

The current state information is available in the FSM_STATUS_DIG_OUT radio register accessible by direct APB access through RRM register map (see RRM registers list).

3.2 Radio FSM interrupts

The Radio FSM provides a dedicated interrupt output signal to the system.

The interrupts can be enabled/disabled individually through the radio controller APB registers:

- Enable/disable through RADIO_CONTROL_IRQ_ENABLE register
- Reading the RADIO_CONTROL_IRQ_STATUS register returns the interrupts status
- Writing a ‘1’ to the RADIO_CONTROL_IRQ_STATUS[x] clears the associated interrupt flag.

See Section 4.3 Radio controller registers for more details.

4 Radio controller

The radio controller is a small block in charge of two features:

- Slow clock period measurement
- Radio FSM interrupt management

4.1 Slow clock measurement

The Radio controller contains a block dedicated to the slow clock measurement.

This measurement:

- is launched automatically by the hardware when the system clock switches on accurate clock (external XO + RC64MPLL mode locked).
- can be launched by the software when needed (by writing zero in CLK32K_PERIOD register)

The result provided by this block is both a period and a frequency information (in two separate results registers). The software can program the window of measurement (in slow clock cycle number) and period result is provided in 16 MHz half-period unit.

4.2 Radio FSM interrupt management

During the sequences, the Radio FSM generates some interruptions to monitor some unexpected behavior at analog level. As the radio FSM block does not have any APB interface, the interrupt control and status flags are managed inside the radio controller block through APB registers:

- RADIO_CONTROL_IRQ_ENABLE register to enable the wanted interrupt sources.
- RADIO_CONTROL_IRQ_STATUS register to get the status (on read) and to clear the interrupt (by writing '1' on the associated bit).

4.3 Radio controller registers

Table 86. Radio Controller registers list

Address offset	Name	RW	Reset	Description
0x00	RADIO_CONTROL_ID	R	0x00001000	Radio controller ID register
0x04	CLK32COUNT_REG	RW	0x00000017	Window length register
0x08	CLK32PERIOD_REG	R	0x00000000	Slow clock period register
0x0C	CLK32FREQUENCY_REG	R	0x00000000	Slow clock frequency register
0x10	RADIO_CONTROL_IRQ_STATUS	RW	0x00000000	Radio controller interrupt status register
0x14	RADIO_CONTROL_IRQ_ENABLE	RW	0x00000000	Radio controller interrupt control register

Table 87. RADIO_CONTROL_ID register description

Bit	Field name	Reset	RW	Description
31:0	IDENTIFICATION	0x1000	R	Radio control Identification register.

Table 88. CLK32COUNT_REG register description

Bit	Field name	Reset	RW	Description
8:0	SLOW_COUNT	0x17	RW	Program the window length (in slow clock period) for slow clock measurement. Slow clock is measured in a window of SLOW_COUNT+1 slow clock cycles.
31:9	RESERVED31_9	0x0	R	Reserved

Table 89. CLK32PERIOD_REG register description

Bit	Field name	Reset	RW	Description
18:0	SLOW_PERIOD	0x0	R	Indicates slow clock period information. The result provided in this field corresponds to the length of SLOW_COUNT periods of the slow clock (32 kHz) measured in 16 MHz half-period unit. Example: if SLOW_COUNT=0x17=23d and SLOW_PERIOD=24000d -> slow clock period = SLOW_PERIOD / (16e6 x 2 x (SLOW_COUNT+1)) = 24000 / (16e6 x 2 x 24) = 31.25e-6 A new calculation can be launched by writing zero in CLK32PERIOD register. In this case, the time window uses the value programmed in SLOW_COUNT field.
31:19	RESERVED31_19	0x0	R	Reserved

Table 90. CLK32FREQUENCY_REG register description

Bit	Field name	Reset	RW	Description
26:0	SLOW_FREQUENCY	0x0	R	Value equal to (239 / SLOW_PERIOD). Warning: This register is updated only 28 x 16 MHz cycles = 1.75 us after the associated IRQ line/status bit are raised.
31:27	RESERVED31_27	0x0	R	Reserved

Table 91. RADIO_CONTROL_IRQ_STATUS register description

Bit	Field name	Reset	RW	Description
0	SLOW_CLK_IRQ	0x0	RW	Slow clock measurement end of calculation interrupt status When read: - 0: no pending interrupt - 1: pending interrupt: slow clock period/frequency values are available. Write '1' to clear the interrupt.
7:1	RESERVED7_1	0x0	R	Reserved
13:8	RADIO_FSM_IRQ	0x0	RW	Radio FSM interrupt status (aka RfFsm_event_irq). -0: no pending interrupt -1: pending interrupt Write '1' to clear the interrupt.
31:14	RESERVED31_14	0x0	R	Reserved

Table 92. RADIO_CONTROL_IRQ_ENABLE register description

Bit	Field name	Reset	RW	Description
0	SLOW_CLK_IRQ_MASK	0x0	RW	Mask slow clock measurement interrupt 0: IT disabled / 1: IT enabled
7:1	RESERVED7_1	0x0	R	Reserved
13:8	RADIO_FSM_IRQ_MASK	0x0	RW	Mask for each RfFsm_event (Radio FSM) interrupt. - 0: Interrupt disabled - 1: Interrupt enabled. RfFsm_event [5] = synth_cal_error RfFsm_event [4] = lock_failed RfFsm_event [3] = synth_unlock_detect RfFsm_event [2] = synth_cal_timeout RfFsm_event [1] = cbp_cal_timeout RfFsm_event [0] = lock_timeout
31:14	RESERVED31_14	0x0	R	Reserved

5 BLE controller sequence

The BLE controller needs a trigger event to start any action. Then the sequencer manages a transmission or reception (or no) sequence depending on the RAM table content it reads.

5.1 Timers

Three different timers can trig the BLE controller sequence:

1. Wake-up timer event
- the event comes from the wake-up timer
- this timer is based on absolute time
- This timer is located in the wake-up block and is the only one able to wake up the BLE IP (and the SoC) from a low power state.

2. Timer1 timer
- the event comes from the Timer1
- this timer uses the interpolated time provided by the wake-up block
- the Timer1 is in fact a comparator between the interpolated time provided by the wake-up block and a match value located in the sequencer of the BLE link layer. It cannot be used in low power mode.

3. Timer2 timer
- the event comes from the Timer2
- this timer is based on a relative time and starts counting at the end of the previous transmission/reception for the duration programmed in Timer2[19:0] field. In particular, if enabled through the TxRxPack RAM table, then it starts counting at the end of the transfer to trig the next RF transaction or if enabled through the TimeoutDestReg APB register, it really starts counting on the next end of Rx/TX that occurs in the radio.
- the Timer2 is a counter located inside the sequencer of the BLE link layer. It cannot be used in low power mode and is supposed to be used for a short time between two Bluetooth transfers.

Each timer is one-shot. This means once it expires, it stops and the software has to reprogram/re-enable it for a new sequence. The three timers are managed in different ways. Therefore, the software has to ensure it does not start a timer while another one is already on-going for the next sequence.

Here is how the three timers are enabled/disabled:
- the wake-up timer is programmed through the wake-up BLUE_WAKEUP_TIME APB register and enabled/ disabled through BLUE_SLEEP_REQUEST_MODE APB register bit 30 (BLE_WAKEUP_EN). This bit is used to unmask the check BLUE-WAKEUP_TIME[31:4] versus ABSOLUTE_TIME[31:4] to generate a wake-up event when the absolute time counter matches with BLUE_WAKEUP_TIME value. It is cleared by HW when the timeout event triggers or it can be cleared through APB to disable the timer before it expires. It has no interference with the two others.
- the Timer1:
  - duration is programmed only through the BLE controller TimeoutDestReg APB register
  - enable is done only through the BLE controller TimeoutDestReg APB register
  - disable can be done through the BLE controller TimeoutDestReg APB register.
- the Timer2:
  - can be programmed either through the BLE controller TimeoutDestReg APB register or through the TxRxPack table
  - can be enabled/disabled either through the BLE controller TimeoutDestReg APB register or through the TxRxPack table.

Note:
- Programming respectively Timer1 or Timer2 through the BLE controller TimeoutDestReg APB register automatically disables respectively the Timer2 or Timer1.
- During sequence execution, Timer1 is disabled when the sequencer treats the enable/disable action on Timer2 through the TxRxPack RAM table whatever the Timer2En bit value.

- If the BLE sequence ends on a receive timeout (StatusReg.RcvTimeout = 1), Timer2 is not started even if the TxRxPack.Timer2En associated to this reception was 1.
- Even if Timer2 can be enabled through the BLE controller TimeoutDestReg APB register, it is recommended not to do it in application flow and to use RAM table.
- TimeoutDestReg[1:0] and TimeoutReg[31:0] need to be programmed at least 15 microseconds before the required start trigger.

5.2 BLE sequence description

The first RAM access done by the sequencer on any trigger event is to get the GlobalStatMach word 0x01 to check the active bit.

If the active bit is low, then nothing is done except:

- setting NoActiveLError flag in the StatusReg BLE APB register
- and if IntNoActiveLError is set in the GlobalStatMach, setting the NoActiveLError in Interrupt1Reg BLE APB register and generating an associated interrupt.

Otherwise, if active bit is high, the parameters that the controller reads during the first phase are considered as ready and updated. Then, the sequencer block starts a sequence divided in several steps.

5.2.1 First initialization step

The 1st INIT step starts on the trigger event (from the wake-up timer or the Timer1 or the Timer2).

During this initialization step, the sequencer only reads the minimum number of parameters to request the radio initialization for a transmission or a reception.

This first initialization step ends on a timeout defined by a bit field in the GlobalStatMach:

- WakeupInitDelay (time unit is 16 x slow clock so typically 512 kHz) when trig event source is the wake-up timer
- Timer12InitDelayCal (time unit is 1 us) when the trig event is the Timer1 or when the trig event is the Timer2 and CalReq bit in TxRxPack table is set (PLL calibration requested)
- Timer2InitDelayNoCal (time unit is 1 us) when the trig event is the Timer2 and CalReq bit in the TxRxPack table is low (no PLL calibration requested).

Note: The BLE wake-up event is based on 32 kHz granularity (absolute_time[31:4]). Then BLE wake-up event occurs at BLUE_WAKEUP_TIME[31:4], but the controller waits until BLUE_WAKEUP_TIME[31:0] + WakeupInitDelay. It means that in any case the sequencer during 1st INIT step manages the 512 kHz granularity.

InitDelay is used as a generic name for this duration to simplify the documentation as it can be three different bit fields that define it depending on the configuration.

When the timeout expires, the sequencer checks several conditions to decide if it switches to the second initialization step or exits with error. The checked conditions are:

- Radio FSM reaches ACTIVE2 state meaning it is ready to receive a TX or RX request (and that system clock is the accurate clock),
- All RAM accesses and radio register writings to be done by the sequencer during the first initialization step are over
- No configuration error has occurred.

If all the conditions are true, then the sequencer:

- sends a TX or RX request to the Radio FSM depending on the transfer direction indicated by TxMode bit in the current StatMach table
- and switches to the second initialization step.

If at least one of the conditions is false then:

- the sequence rises the flag(s) associated to the error(s). It can be:
  - StatusReg.ConfigError bit if a configuration error has been detected
  - StatusReg.Active2Error bit if the Radio FSM is not in ACTIVE2 at the end of the InitDelay
  - StatusReg.SemaTimeoutError bit if the semaphore did not grant the BLE IP on time
  - DebugStatusReg.SeqEerror[2] bit if the sequencer did not finish all AHB read/write accesses planned during the first initialization step when timeout occurs.
- No RAM write back action is done.

- The error bits set in StatusReg also appear in Interrupt1Reg if their associated interrupt enable bit is set in the GlobalStatMach table.

5.2.2 Second Initialization step

The 2nd INIT step starts when the TX or RX request is sent to the radio FSM and once few delays have been read in the GlobalStatMach by the sequencer. Those delays are needed during the 2nd INIT and DATA INIT steps.

TxRxPack.AllTableReady is the first flag that the sequencer checks during the second initialization step: if it reads TxRxPack.AllTableReady = 1 then everything is OK and it continues reading the remaining parameters required for the second initialization step. If the sequencer reads TxRxPack.AllTableReady = 0 then this means that the RAM table programming in not coherent and the sequencer stops its sequence by sending an interrupt to the CPU (if enabled) and sets StatusReg.AllTableReadyError = 1.

During the 2nd INIT step, the sequencer gets all the information from the RAM tables linked to the transfer to proceed (except DatPtr and TxDataReady bit fields).

This means the software must have filled all the RAM tables information (except DatPtr and TxDataReady bit fields) when the InitDelay timeout expires.

This 2nd INIT step ends on a timeout based on 2-bit fields in the GlobalStatMach:

- init_radio_delay is used as a generic name for this duration to simplify the documentation as it can come from 4 different bit fields depending on the transfer configuration:
  - TransmitNoCalDelayChk when the transfer is a TX and no PLL calibration is requested (CalReq bit is low),
  - TransmitCalDelayChk when the transfer is a TX and a PLL calibration is requested (CalReq bit is set),
  - ReceiveNoCalDelayChk when the transfer is an RX and no PLL calibration is requested (CalReq bit is low),
  - ReceiveCalDelayChk when the transfer is an RX and a PLL calibration is requested (CalReq bit is set),
- TxdataReadyCheck

The init_radio_delay duration must not exceed the RF analog set-up time up to power on the antenna for a transmission (or ready to receive on the antenna). This means it must not exceed:

- the duration of the radio FSM to go from ACTIVE2 to TX state for a transmission with few us of margin
- the duration of the radio FSM to go from ACTIVE2 to RX state for a reception.

Note: For transmission, the init_radio_delay timeout must expire before the radio FSM is in TX mode to avoid missing the start of the preamble sending on the antenna (or else garbage is sent while preamble is supposed to be output). For a reception, the init_radio_delay must expire close to the switch in RX state of the Radio FSM, knowing the RcvTimeout count starts when the init_radio_delay expires.

At very beginning of the 2nd INIT step:

- the sequencer starts an internal relative timer
- in parallel, the sequencer reads the init_radio_delay, ConfigEndDuration and TxdataReadyCheck information in the GlobalStatMach.

The GlobalStatMach.ConfigEndDuration bit field allows delaying the reading of the transfer information contained in the RAM tables by the sequencer. Indeed, the init_radio_delay (2nd INIT + DATA INIT steps duration) must fit in the analog radio set-up duration which is supposed to be longer than the RAM tables reading by the sequencer. The 2nd INIT ends when the relative timer reaches “init_radio_delay – TxdataReadyCheck”.

5.2.3 Data initialization step

This Data INIT step starts when the 2nd INIT step ends.

- During this step, the sequencer only gets 2 values from the table
- TxDataReady bit in the TxRxPack indicating enough bytes are present in the TX payload data buffer (in case of transmission only). The CPU has to set TxRxPack.TxdataReady = 1 when at least four 32-bit words are available in the transmission buffer. TxRxPack.TxdataReady is the last parameter that the sequencer reads before starting to prefetch the data payload: if it reads TxRxPack.TxdataReady = 1 then everything is OK and it continues prefetching the data payload to fill its internal FIFOs. If the sequencer reads TxRxPack.TxdataReady = 0 then this means that the minimum data buffer filling is not enough and the sequencer stops its sequence by sending an interrupt to the CPU (if enabled) and set StatusReg.TxDataReadyError = 1.
- DatPtr bit field in the TxRxPack.

The GlobalStatMach.TxdataReadyCheck is used to delay the start of this DATA INIT step to allow more time to the software to provide the data pointer (and first values to transmit if transfer is a transmission).

The DATA INIT step ends when the relative timer reaches init_radio_delay:

- if all conditions are OK AllTableReady read at 1, TxDataReady read at 1 for a transmission, radio FSM still in a state between ACTIVE2 and RX or TX, a start pulse is sent to the receive/transmit block for a reception/transmission.
- or else no start pulse is sent to the receive/transmit block and status/interrupt flags are updated in the BLE APB registers (no RAM write back is done).

The transmit block receives some information from the radio FSM, it is in TX state to synchronize the moment data can start to be sent to the modulator. As the transmit block is supposed to receive the start pulse from the sequencer a bit before the radio FSM reaches the TX state, a wait window needs to be defined to avoid waiting forever: this time window is defined in the GlobalStatMach.TxReadyTimeout bit field.

Caution: It is the responsibility of the software to ensure that the init_radio_delay, the ConfigEndDuration and the TxdataReadyCheck values are coherent to guarantee both data ready on time in the table and start pulse sent on time to the receive/transmit block.

5.2.4 Transmission/reception step

The transmission / reception step starts when the start pulse is sent by the sequencer to the transmit or to the receive block.

This step ends when the transmit/receive block indicates the transfer is done:

- all data transmitted for a transmission (followed by a waiting time defined by GlobalStatMach.TxdelayEnd)
- a frame has been received or the programmed timeout to wait for a reception expired without any reception.

Important:

- When a transmission is completed, the timer2, if it is programmed, starts counting only when GlobalStatMach.TxdelayEnd is elapsed
- When a reception is completed, if the exit reason is a timeout the timer2 does not start.

5.2.5 Context saving step

Once the sequence receives the tx_done pulse from the transmit block or the rcv_done pulse from the receive block, it starts the RAM write back operation.

The RAM write back impacts the following RAM table elements:

- GlobalStatMach Word1 if the GlobalStatMach.ChkFlagAutoClearEna bit is set:
  - clear the Active bit
  - write back the rest of the bit field of this word1 with value previously read by the sequencer in the RAM table.
- StatMach Word0
- StatMach Word1: update the TxPoint[31:0] with TxPointNext[31:0] or keep the same.
- StatMach Word2: update the RcvPoint[31:0] with RcvPointNext[31:0] or keep the same.
- StatMach Word3: update the TxPointPrev[31:0] with TxPoint[31:0] or keep the same.
- StatMach Word4: update the RcvPointPrev[31:0] with RcvPoint[31:0] or keep the same.
- StatMach Word5: update the TxPointNext[31:0] or keep the same.
- StatMach Word6: update the PCntTx[31:0] or keep the same.
- StatMach Word7: update the PCntTx[39:32] and PCntRcv[23:16] or keep the same.
- StatMach Word8:
  - update the PCntRcv[39:24] or keep the same
  - the rest of the Word8 is written back with value previously read by the sequencer in the RAM table.

5.3 BLE sequence summary

The sequences of operations characterizing a transmission and a reception are summarized in the following timing diagrams.

Figure 1. **TX sequence**

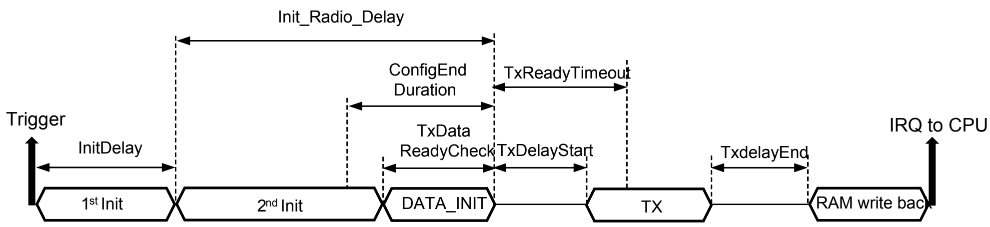
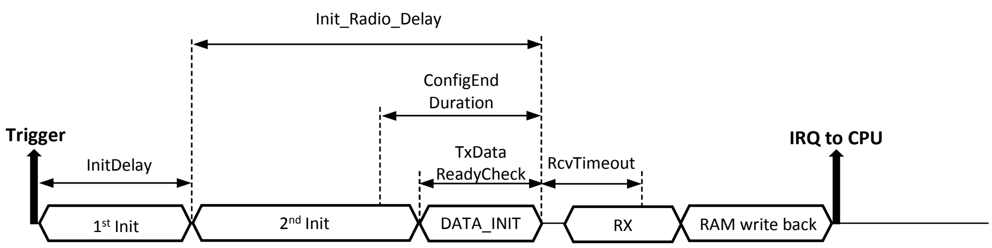


Figure 2. **RX sequence**



5.4 TX and RX sequence signals

The controller is able to control an external power amplifier (located on the board to increase the TX power) through a signal provided to the SoC to be connected directly on an external GPIO or to be managed with additional logical mechanism. The external PA can be useful to extend the TX power.

This signal is raised as soon as a transmission is requested (2^{nd} INIT step starting point) and goes down as soon as the system switches off the internal PA.

On the other hand, the controller provides a signal to monitor a reception request.

See TX_SEQUENCE and RX_SEQUENCE GPIO alternate functions on the BlueNRG-LP reference manual (RM0479) to see what pins support this feature.

5.5 BLE controller registers

Table 93. BLE controller register list

Address offset	Name	RW	Reset	Description
0x00	RESERVED	R	0x00010100	Reserved.
0x04	INTERRUPT1REG	RW	0x00000000	Interrupt pending and interrupt clear register 1.
0x08	INTERRUPT2REG	RW	0x00000000	Interrupt pending and interrupt clear register 2.
0x0C	TIMEOUTDESTREG	RW	0x00000000	Timer1 and Timer2 enable/disable.
0x10	TIMEOUTREG	RW	0x00000000	Timer1 and Timer2 timeout register.
0x14	TIMERCAPTUREREG	R	0x00000000	Timer capture register.
0x18	CMDREG	RW	0x00000000	CmdReg register.
0x1C	STATUSREG	R	0x00000000	Status register.
0x20	INTERRUPT1ENABLEREG	R	0x00000000	This read-only register is a copy/summary of all the enable mask bits located in the different RAM tables. When ‘0’, corresponding interrupt was masked during previous sequence. When ‘1’, corresponding interrupt was enabled during the previous sequence.
0x24	INTERRUPT1LATENCYREG	R	0x00000000	Interrupt1 Latency register.
0x28	MANAESKEY0REG	RW	0x00000000	Manual AES Key0 register
0x2C	MANAESKEY1REG	RW	0x00000000	Manual AES Key1 register
0x30	MANAESKEY2REG	RW	0x00000000	Manual AES Key2 register
0x34	MANAESKEY3REG	RW	0x00000000	Manual AES Key3 register
0x38	MANAESCLEARTEXT0REG	RW	0x00000000	Manual AES ClearText0 register
0x3C	MANAESCLEARTEXT1REG	RW	0x00000000	Manual AES ClearText1 register
0x40	MANAESCLEARTEXT2REG	RW	0x00000000	Manual AES ClearText2 register
0x44	MANAESCLEARTEXT3REG	RW	0x00000000	Manual AES ClearText3 register
0x48	MANAESCIPHERTEXT0REG	R	0x00000000	Manual AES CipherText0 register
0x4C	MANAESCIPHERTEXT1REG	R	0x00000000	Manual AES CipherText1 register
0x50	MANAESCIPHERTEXT2REG	R	0x00000000	Manual AES CipherText2 register
0x54	MANAESCIPHERTEXT3REG	R	0x00000000	Manual AES CipherText3 register
0x58	MANAESCMDREG	RW	0x00000000	Manual AES CmdReg register
0x5C	MANAESSTATREG	R	0x00000000	Manual AES Status register
0x60	AESLEPRIVPOINTERREG	RW	0x00000000	AES LE Privacy Pointer register
0x64	AESLEPRIVHASHREG	RW	0x00000000	AES LE Privacy Hash register
0x68	AESLEPRIVPRANDREG	RW	0x00000000	AES LE Privacy Prand register
0x6C	AESLEPRIVCMDREG	RW	0x00000000	AES LE Privacy CmdReg register
0x70	AESLEPRIVSTATREG	R	0x00000000	AES LE Privacy Status register
0x74	DEBUGCMDREG	RW	0x00000000	DebugCmdReg register
0x78	DEBUGSTATUSREG	R	0x00000000	DebugStatusReg register

Table 94. INTERRUPT 1REG register description

Bit	Field name	Reset	RW	Description
3:0	RESERVED3_0	0x0	R	Reserved
4	ADDPOINTERROR	0x0	RW	Address Pointer Error. When read, indicates the interrupt status. Write 1'b1 to clear.
5	RXOVERFLOWERROR	0x0	RW	Receive Overflow. When read, indicates the interrupt status. Write 1'b1 to clear.
6	RESERVED6	0x0	R	Reserved
7	SEQDONE	0x0	RW	Sequencer end of task. When read, indicates the interrupt status. Write 1'b1 to clear.
8	TXERROR_0	0x0	RW	Transmission error 0 AES did not acknowledge the transmit block request on time. When read, indicates the interrupt status. Write 1'b1 to clear. Note: On this error, the transmit block stops the on-going transmission but the sequencer manages it as a normal end of transmission.
9	TXERROR_1	0x0	RW	Transmission error 1: a TX skip happened during an on-going transmission. When read, indicates the interrupt status. Write 1'b1 to clear.
10	TXERROR_2	0x0	RW	Transmission error 2: channel index is greater than 39. When read, indicates the interrupt status. Write 1'b1 to clear.
11	TXERROR_3	0x0	RW	Transmission error 3: Radio FSM did not provide the tx_ready information on time (timeout defined in GlobalStatMach.TxReadyTimeout[7:0] bit field). When read, indicates the interrupt status. Write 1'b1 to clear.
12	RESERVED	0x0	RW	Reserved.
13	ENCERROR	0x0	RW	Encryption error on receive. When read, indicates the interrupt status. Write 1'b1 to clear.
14	ALLTABLEREADYERROR	0x0	RW	All RAM Table not ready on time. When read, indicates the interrupt status. Write 1'b1 to clear.
15	TXDATAREADYERROR	0x0	RW	Transmit data pack (TxRxDat) not ready when TX on antenna was about to start. When read, indicates the interrupt status. Write 1'b1 to clear.
16	NOACTIVELERROR	0x0	RW	GlobStatMach.active bit error (read as 0 on a trigged sequence). When read, indicates the interrupt status. Write 1'b1 to clear.
17	RESERVED	0x0	RW	Reserved.
18	RCVLENGTHERROR	0x0	RW	Receive length error. When read, indicates the interrupt status.

Bit	Field name	Reset	RW	Description
				Write 1'b1 to clear.
19	SEMATIMEOUTERROR	0x0	RW	Semaphore timeout error. When read, indicates the interrupt. Write 1'b1 to clear.
20	RESERVED	0x0	RW	Reserved.
21	TXRXSKIP	0x0	RW	Transmission/Reception skip. When read, indicates the interrupt status. Write 1'b1 to clear.
22	ACTIVE2ERROR	0x0	RW	Active2 Radio state error. When read, indicates the interrupt status. Write 1'b1 to clear.
23	CONFIGERROR	0x0	RW	Data pointer configuration error. When read, indicates the interrupt status. Write 1'b1 to clear.
24	TXOK	0x0	RW	Previous transmitted packet received OK by the peer device. When read, indicates the interrupt status. Write 1'b1 to clear.
25	DONE	0x0	RW	Receive/Transmit done. When read, indicates the interrupt status. Write 1'b1 to clear.
26	RCVTIMEOUT	0x0	RW	Receive timeout (no preamble found). When read, indicates the interrupt status. Write 1'b1 to clear.
27	RCVNOMD	0x0	RW	Received MD bit embedded in the PDU data packet header was zero. When read, indicates the interrupt status. Write 1'b1 to clear.
28	RCVCMD	0x0	RW	Received command. When read, indicates the interrupt status. Write 1'b1 to clear.
29	TIMECAPTURETRIG	0x0	RW	TimerCaptureReg time capture. When read, indicates the interrupt status. Write 1'b1 to clear.
30	RCVCRCERR	0x0	RW	Receive data fail (CRC error). When read, indicates the interrupt status. Write 1'b1 to clear. Note: This error is raised only if at least preamble and access address have been detected.
31	RCVOK	0x0	RW	Receive data OK. When read, indicates the interrupt status. Write 1'b1 to clear.

Table 95. **INTERRUPT2REG register description**

Bit	Field name	Reset	RW	Description
0	AESMANENCINT	0x0	RW	AES manual encryption. When read, indicates the interrupt status. Write 1'b1 to clear.
1	AESLEPRIVINT	0x0	RW	AES LE privacy engine. When read, indicates the interrupt status. Write 1'b1 to clear.
31:2	RESERVED31_2	0x0	R	Reserved.

Table 96. **TIMEOUTDESTREG register description**

Bit	Field name	Reset	RW	Description
1:0	DESTINATION	0x0	RW	Timeout timer Destination - 00 or 01: all disabled - 10: Timer1 enable - 11: Timer2 enable (but Timer2 really starts counting at the end of a Rx/Tx sequence) Note: Enabling one of the two timers automatically disables the second one.
31:2	RESERVED31_2	0x0	R	Reserved

Table 97. **TIMEOUTREG register description**

Bit	Field name	Reset	RW	Description
31:0	TIMEOUT	0x0	RW	Timer1 or Timer2 Timeout value (depending on destination register). Time units: - in microseconds for Timer2 - in periods of 512 kHz clock for Timer1.

Table 98. **TIMERCAPTUREREG register description**

Bit	Field name	Reset	RW	Description
31:0	TIMERCAPTURE	0x0	R	Interpolated absolute time capture register (cf. TxRxPack.TrigRcv/TrigDone, GlobStatMach.TimeCapture/TimeCaptureSel for detailed specifications) This register is cleared on the beginning of a new BLE sequence. Time unit is in 16 x slow clock so typically 512 kHz period cycle.

Table 99. CMDREG register description

Bit	Field name	Reset	RW	Description
0	TXRXSKIP	0x0	R	Transmission/Reception skip command. This bit is auto-cleared by the HW.
1:2	RESERVED	0x0	RW	Reserved.
3	CLEARSEMAREQ	0x0	R	Semaphore Clear command. Setting this bit releases the token for the IP_BLE. Software option in parallel with the hardware management by the BLE sequencer through TxRxPack.KeepSemaReq bit. This bit is auto-cleared by the HW.
31:4	RESERVED31_4	0x0	R	Reserved

Table 100. **STATUSREG register description**

Bit	Field name	Reset	RW	Description
3:1	RESERVED3_1	0x0	R	Reserved
4	ADDPOINTERROR	0x0	R	Address Pointer Error status.
5	RXOVERFLOWERROR	0x0	R	AHB arbiter is full and there is no more storage capability available in RX data path
6	PREVTRANSMIT (*)	0x0	R	Previous event was a Transmission (1) or Reception (0) status
7	SEQDONE	0x0	R	Sequencer end of task status. This bit is set each time the sequencer ends the execution of a sequence due to a trigger event whatever the result (OK, with errors, ACTIVE bit not set, etc.).
8	TXERROR_0	0x0	R	Transmission error 0 status Transmit block missing data error (when the transmit block has to transmit serially a bit and has no more data as the AES block did not provide a new data byte yet). Note: On this error, the transmit block stops the on-going transmission but the sequencer manages it as a normal end of transmission. The flag is the only information available for the user.
9	TXERROR_1	0x0	R	Transmission error 1 status (when CmdReg.TxRxSkip=1 happens when a transmission is on-going).
10	TXERROR_2	0x0	R	Transmission error 2 status (if StateMach.Remap_chan >39).
11	TXERROR_3	0x0	R	Transmission error 3 status (if i_tx_ready=0 after the time value defined by GlobStatMach.TxReadyTimeout.)
12	TXERROR_4	0x0	R	Transmission error 4 status (the SupplementalTime field is not between 2 and 20 inclusive or in case of coded packet or if Supplemental type is 3)
13	ENCERROR	0x0	R	Encryption error on receive status
14	ALLTABLEREADYERROR	0x0	R	All RAM Table not ready status
15	TXDATAREADYERROR	0x0	R	Transmit data pack (TxRxDat) not ready status
16	NOACTIVELERROR	0x0	R	GlobStatMach.active bit error (read as 0) status
17	RESERVED	0x0	R	(Was previously INITDELAYERROR but this error can no longer occur with new Time Interpolator implementation)
18	RCVLENGTHERROR	0x0	R	Receive length error status
19	SEMATIMEOUTERROR	0x0	R	Semaphore timeout error status
20	SEMAWASPREEMPT (*)	0x0	R	BLE has been preempted. Semaphore status
21	TXRXSKIP	0x0	R	Transmission/Reception skip status
22	ACTIVE2ERROR	0x0	R	Active2 Radio state error status
23	CONFIGERROR (*)	0x0	R	Data pointer configuration error status
24	TXOK	0x0	R	Previous transmitted packet received OK by the peer device status. This bit is updated at the end of a reception. 0: the previous transmitted packet was not received OK by the peer device. 1: the previous transmitted packet was received OK by the peer device. This bit is set only if the following conditions are verified: - this is a data packet, - the SN/NESN mechanism is enabled (TxRxPack.SN_EN = 1), - a preamble and a good access address have been received inside the receive window, - the received NESN is different from the local StatMach.SN bit.
25	DONE	0x0	R	Receive/Transmit done status. This flag is set if the sequencer reached the TX/RX phase (start pulse sent to the receive/ transmit block which returned a done pulse).

Bit	Field name	Reset	RW	Description
26	RCVTIMEOUT	0x0	R	Receive timeout status (no preamble found).
27	RCVNOMD	0x0	R	Received MD bit embedded in the PDU data packet header was zero status
28	RCVCMD	0x0	R	Received command status
29	TIMECAPTURETRIG	0x0	R	TimerCaptureReg time capture status
30	RCVCRCERR	0x0	R	Receive data fail. (CRC error or preamble not found) status
31	RCVOK	0x0	R	Receive data OK status

Note:
- *This StatusReg is updated on each BLE sequencer end of sequence.*
- *This register is cleared each time the BLE sequencer starts a new sequence (timer trig event) except for the bit tagged with (*):*
  - *CONFIGERROR: updated when the sequencer reads the StatMach*
  - *PREVTRANSMIT: updated when the sequencer reaches the TX/RX step again*
- *After a reception, an SN_NESN error is identified if the StatusReg indicates the RX is done (DONE=1), not OK (RCVOK=0) but no specific error flag is set.*
- *When a proper transmission occurred, the DONE flag (in StatusReg and potentially Interrupt1Reg) and the StatusReg.PrevTransmit bit are set.*

Table 101. **INTERRUPT1ENABLEREG register description**

Bit	Field name	Reset	RW	Description
3:0	RESERVED3_0	0x0	R	Reserved.
4	ADDPOINTERROR	0x0	R	Address Pointer Error enable interruption (Read Only).
5	RXOVERFLOWERROR	0x0	R	RX Overflow Error enable interruption (Read Only).
6	RESERVED6	0x0	R	Reserved.
7	SEQDONE	0x0	R	Sequencer end of task enable interruption (Read Only).
8	TXERROR_0	0x0	R	Transmission error 0 enable interruption (Read Only).
9	TXERROR_1	0x0	R	Transmission error 1 enable interruption (Read Only).
10	TXERROR_2	0x0	R	Transmission error 2 enable interruption (Read Only).
11	TXERROR_3	0x0	R	Transmission error 3 enable interruption (Read Only)
12	RESERVED	0x0	R	Reserved.
13	ENCERROR	0x0	R	Encryption error on receive enable interruption (Read Only).
14	ALLTABLEREADYERROR	0x0	R	All RAM Table not ready enable interruption (Read Only).
15	TXDATAREADYERROR	0x0	R	Transmit data pack (TxRxDat) not ready enable interruption (Read Only).
16	NOACTIVELERROR	0x0	R	GlobStatMach.active bit error (read as 0) enable interruption (Read Only).
17	RESERVED	0x0	R	Reserved.
18	RCVLENGTHERROR	0x0	R	Receive length error enable interruption (Read Only)
19	SEMATIMEOUTERROR	0x0	R	Semaphore timeout error enable interruption (Read Only).
20	RESERVED	0x0	R	Reserved.
21	TXRXSKIP	0x0	R	Transmission/Reception skip enable interruption (Read Only).
22	ACTIVE2ERROR	0x0	R	Active2 Radio state error enable interruption (Read Only).
23	CONFIGERROR	0x0	R	Data pointer configuration error enable interruption (Read Only).
24	TXOK	0x0	R	Previous transmitted packet received OK enable interruption (Read Only).
25	DONE	0x0	R	Receive/Transmit done interruption (Read Only).
26	RCVTIMEOUT	0x0	R	Receive timeout enable interruption (Read Only) (no preamble found).
27	RCVNOMD	0x0	R	Received MD bit embedded in the PDU data packet header was zero enable interruption (Read Only).
28	RCVCMD	0x0	R	Received command enable interruption (Read Only).
29	TIMECAPTURETRIG	0x0	R	TimerCaptureReg time capture enable interruption (Read Only).
30	RCVCRCERR	0x0	R	Receive data fail enable interruption (Read Only).
31	RCVOK	0x0	R	Receive data OK enable interruption. (Read Only).

Table 102. **INTERRUPT1LATENCYREG register description**

Bit	Field name	Reset	RW	Description
7:0	INTERRUPT1LATENCY	0x0	R	Relative time counter after interrupt1. - Time unit: 1us - Clamped at 255. Reset when all interrupt1 sources are cleared or when a new interrupt1 IRQ is raised.
31:8	RESERVED31_8	0x0	R	Reserved

Table 103. MANAESKEY0REG register description

Bit	Field name	Reset	RW	Description
31:0	MANAESKEY_31_0	0x0	RW	Manual mode AES key

Table 104. MANAESKEY1REG register description

Bit	Field name	Reset	RW	Description
31:0	MANAESKEY_63_32	0x0	RW	Manual mode AES key

Table 105. MANAESKEY2REG register description

Bit	Field name	Reset	RW	Description
31:0	MANAESKEY_95_64	0x0	RW	Manual mode AES key

Table 106. MANAESKEY3REG register description

Bit	Field name	Reset	RW	Description
31:0	MANAESKEY_127_96	0x0	RW	Manual mode AES key

Table 107. MANAESCLEARTEXT0REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CLEAR_31_0	0x0	RW	Manual AES Clear Text

Table 108. MANAESCLEARTEXT1REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CLEAR_63_32	0x0	RW	Manual AES Clear Text

Table 109. MANAESCLEARTEXT2REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CLEAR_95_64	0x0	RW	Manual AES Clear Text

Table 110. MANAESCLEARTEXT3REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CLEAR_127_96	0x0	RW	Manual AES Clear Text

Table 111. MANAESCHIPHERTEXT0REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CIPHER_31_0	0x0	R	Manual AES cipher text

Table 112. MANAESCHIPHERTEXT1REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CIPHER_63_32	0x0	R	Manual AES cipher text

Table 113. MANAESCHIPHERTEXT2REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CIPHER_95_64	0x0	R	Manual AES Cipher Text

Table 114. MANAESCHIPHERTEXT3REG register description

Bit	Field name	Reset	RW	Description
31:0	AES_CIPHER_127_96	0x0	R	Manual AES cipher Text

Table 115. MANAESCMDREG register description

Bit	Field name	Reset	RW	Description
0	START	0x0	R	AES manual encryption Start command. This bit is auto-cleared by the HW.
1	INTENA	0x0	RW	AES manual encryption interrupt enable on Interrupt2Reg.
31:2	RESERVED31_2	0x0	R	Reserved.

Table 116. MANAESSTATREG register description

Bit	Field name	Reset	RW	Description
0	BUSY	0x0	R	AES manual encryption busy status
31:1	RESERVED31_1	0x0	R	Reserved

Table 117. AESLEPRIVPOINTERREG register description

Bit	Field name	Reset	RW	Description
23:0	POINTER	0x0	RW	AES LE privacy pointer
31:24	RESERVED31_24	0x0	R	Reserved

Table 118. AESLEPRIVHASHREG register description

Bit	Field name	Reset	RW	Description
23:0	HASH	0x0	RW	AES LE privacy Reference Hash
31:24	RESERVED31_24	0x0	R	Reserved

Table 119. AESLEPRIVCMDREG register description

Bit	Field name	Reset	RW	Description
0	START	0x0	R	AES LE privacy Start command. This bit is auto-cleared by the HW.
1	INTENA	0x0	RW	AES LE privacy interrupt enable on Interrupt2Reg.
9:2	NBKEYS	0x0	RW	AES LE privacy number of keys pointed by AesLePrivPointerReg (points to the resolution key list).
31:10	RESERVED31_10	0x0	R	Reserved.

Table 120. AESLEPRIVSTATREG register description

Bit	Field name	Reset	RW	Description
0	BUSY	0x0	R	AES LE privacy busy status.
1	KEYFND	0x0	R	AES LE privacy key finding status.
9:2	KEYFNDINDEX	0x0	R	AES LE privacy index of the key found in the resolution key list.
31:10	RESERVED31_10	0x0	R	Reserved.

Table 121. DEBUGCMDREG register description

Bit	Field name	Reset	RW	Description
0	CLEARDEBUGINT	0x0	RW	Debug interrupt. Write '1' to clear.
1	SEQDEBUGMODE	0x0	RW	Enable the debug mode for sequencer.
5:2	SEQDEBUGBUSSEL[30]	0x0	RW	Sequencer debug bus selection for DebugStatusReg[50] meaning.
15:6	RESERVED15_6	0x0	R	For future use.
19:16	AESDEBUGMODE	0x0	RW	AES debug flags clear. All bits must be written to '1' together to clear the DebugStatusReg[19:16] bits. Caution: this bit field is not cleared by hardware. The software must clear it to be able to get again the AES debug flag information. Note: This clear operation is possible/taken into account only while AES is ON.
31:20	RESERVED31_20	0x0	R	Reserved.

Table 122. DEBUGSTATUSREG register description

Bit	Field name	Reset	RW	Description
6:0	DEBUGSTATUSREG[6:0]	0x0	R	Depending on DebugCmdReg[5:2] bit field value - If DebugCmdReg[5:2] = 0 (default): • SEQERROR_4: detect something was wrong during the sequence and no RAM write back occurred (backdoor for debug). • SEQERROR_3: detect when Rcv/Transmit(No)CalDelayChk duration is elapsed, but all required conditions are not fulfilled • SEQERROR_2: detect when InitDelay duration is elapsed, but all required conditions are not fulfilled • SEQERROR_1: detect when (internal) watchdog timer is elapsed, but the Writeback phase is not finished. • SEQERROR_0: detect when a trig event happened but the sequencer was not in IDLE state (if a sequence was currently running). - If DebugCmdReg[5:2] = 1:

Bit	Field name	Reset	RW	Description
				• DebugStatusReg[6:5] = 0 • DebugStatusReg[4:0] = MAIN_STATE[4:0] - If DebugCmdReg[5:2] = 2: • DebugStatusReg[6] = 0 • DebugStatusReg[5:0] = SLV_STATE[4:0] - If DebugCmdReg[5:2] = 3: • DebugStatusReg[6] = 0 • DebugStatusReg[5:0] = SLV_STATE_ERROR[4:0] - If DebugCmdReg[5:2] = others: • DebugStatusReg[6:0] = 0
15:7	RESERVED15_6	0x0	R	For future use.
16	AESDBG_0	0x0	R	AES RX error flag. RX tries to write a data while the packet key is not available. Cleared by writing 0xF in DebugCmdReg[19:16].
17	AESDBG_1	0x0	R	AES TX error flag. TX tries to read a data while the packet key is not available. Cleared by writing 0xF in DebugCmdReg[19:16].
18	AESDBG_2	0x0	R	MIC error flag. Cleared by writing 0xF in DebugCmdReg[19:16]
19	AESDBG_3	0x0	R	For future use
31:20	RESERVED31_20	0x0	R	For future use

6 BLE RAM tables

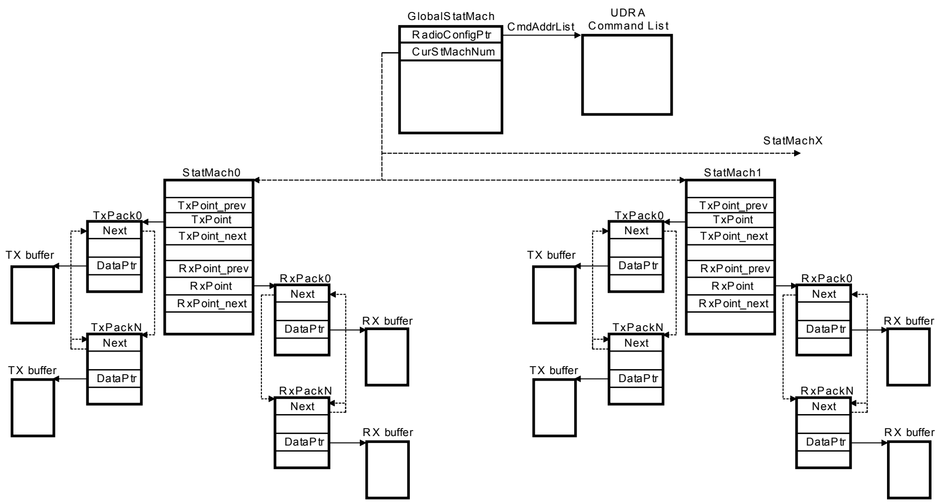
Each time a trigger event is sent to the BLE controller, the sequencer fetches the RAM tables in RAM to get the needed information to know what to configure for the radio and which sequence to start (RX or TX)

There are several types of tables:

- The GlobalStatMach: this table is unique.
- The StatMach: one table by active connection (up to 128 supported by the hardware).
- The TxRxPack: one table packet in RX or in TX. So, there is no predefined number of those tables. They are used as link list from one packet to another during a full connection.
- The DataPack tables corresponding to the data buffers pointed by the DataPtr in the TxRxPack. It contains the PDU section of the Bluetooth packet.

The following figure gives an overview of RAM tables dependencies.

Figure 3. RAM table tree



6.1 GlobalStatMach RAM table

The GlobalStatMach location is frozen by the hardware. This address is 0x200000C0. The GlobalStatMach is unique and mainly contains static information/options.

Table 123. GlobalStatMach

Address offset	Name	RW	Reset	Description
0x00	WORD0	RW	0x00000000	Word0 register
0x04	WORD1	RW	0x00000000	Word1 register
0x08	WORD2	RW	0x00000000	Word2 register
0x0C	WORD3	RW	0x00000000	Word3 register
0x10	WORD4	RW	0x00000000	Word4 register
0x14	WORD5	RW	0x00000000	Word5 register
0x18	WORD6	RW	0x00000000	Word6 register

Table 124. GlobalStatMach.WORD0 register description

Bit	Field name	Reset	RW	Description
31:0	RADIOCONFIGPTR	0x0000000	RW	Radio Configuration address Pointer. Contains the address of the command_start_list used by the RRM block to execute UDRA command. Note: This value is loaded automatically by the RRM when the BLE controller exits reset. However, it is also possible to make the RRM reload it through a reload command in UDRA_CTRL register.

Table 125. GlobalStatMach.WORD1 register description

Bit	Field name	Reset	RW	Description
6:0	CURSTMACHNUM	0x0	RW	Current connection machine number. Defines the state machine number (in the range from 0 to 127) which is running for the current transmission or reception. It is used to calculate the RAM address from which the State machine table ("StateMach") is read. Note: This field is written back with value read at the beginning of the BLE sequence only if the ChkFlagAutoClearEna bit = '1'.
7	ACTIVE	0x0	RW	Must be at '1' when the trig event (Wake-up Timer, Timer1 or Timer2) occurs to start a BLE controller sequence. Otherwise no RF sequence nor timer management is done by the BLE controller. Only SeqDone and NoActiveLError flags are raised in StatusReg and Interrupt1Reg (if associated interrupts are enabled). Note: This field is written back to '0' only if the ChkFlagAutoClearEna bit = '1'.
15:8	WAKEUPINITDELAY	0x0	RW	Delay between wake-up timer trig event on sequencer and RX/TX request sending to the Radio FSM. It corresponds to the sequencer 1st INIT step duration. Note: This bit field is not used if trig event comes from Timer1 or Timer2. The time unit for this delay/value is a period of slow clock frequency x 16 (if slow clock is 32kHz, this bit field unit is 1 period of 512kHz). Note: This field is written back with value read at the beginning of the BLE sequence only if the ChkFlagAutoClearEna bit = '1'.
23:16	TIMER12INITDELAYCAL	0x0	RW	Delay between Timer1 or Timer2 trig event on sequencer and RX/TX request sending to the Radio FSM. It corresponds to the sequencer 1st INIT step duration. Note: This bit field is used for Timer2 trig event only if CalReq bit is set in current TxRxPack RAM table (PLL Calibration is requested. The time unit for this delay is 1us. Note: This field is written back with value read at the beginning of the BLE sequence only if the ChkFlagAutoClearEna bit = '1'.
31:24	TIMER2INITDELAYNOCAL	0x0	RW	Delay between Timer2 trig event on sequencer and RX/TX request sending to the Radio FSM. It corresponds to the sequencer 1st INIT step duration. This bit field is used for Timer2 trig event only if CalReq bit is low in current TxRxPack RAM table (No PLL Calibration is requested). The time unit for this delay is 1us. Note: This field is written back with value read at the beginning of the BLE sequence only if the ChkFlagAutoClearEna bit = '1'.

Table 126. GlobalStatMach.WORD2 register description

Bit	Field name	Reset	RW	Description
7:0	TRANSMITCALDELAYCHK	0x0	RW	Delay between TX request sent to the Radio FSM and the start pulse sent to the transmit block. It corresponds to the sequencer 2^{nd} INIT + DATA INIT steps duration. Note: This bit field is used if TxMode bit is set in the StatMach (transmission) and the CalReq bit is set in current TxRxPack RAM table (PLL Calibration is requested). The time unit for this delay is 1us.
15:8	TRANSMITNOCALDELAYCHK	0x0	RW	Delay between TX request sent to the Radio FSM and the start pulse to the transmit block. It corresponds to the sequencer 2^{nd} INIT + DATA INIT steps duration. Note: This bit field is used if TxMode bit is set in the StatMach (transmission) and the CalReq bit is low in current TxRxPack RAM table (no PLL Calibration is requested). The time unit for this delay is 1us.
23:16	RECEIVECALDELAYCHK	0x0	RW	Delay between RX request sent to the Radio FSM and the start pulse sent to the receive block. It corresponds to the sequencer 2^{nd} INIT + DATA INIT steps duration. Note: This bit field is used if TxMode bit is low in the StatMach (reception) and the CalReq bit is set in current TxRxPack RAM table (PLL Calibration is requested). The time unit for this delay is 1us.
31:24	RECEIVENOCALDELAYCHK	0x0	RW	Delay between RX request sent to the Radio FSM and the start pulse to the receive block. It corresponds to the sequencer 2^{nd} INIT + DATA INIT steps duration. Note: This bit field is used if TxMode bit is low in the StatMach (reception) and the CalReq bit is low in current TxRxPack RAM table (no PLL Calibration is requested). The time unit for this delay is 1us.

Table 127. GlobalStatMach.WORD3 register description

Bit	Field name	Reset	RW	Description
7:0	CONFIGENDDURATION	0x0	RW	Duration for the BLE sequencer to execute the final configuration. The goal of this bit field is to provide more time to the Firmware to prepare the RAM tables. The BLE sequencer waits for relative time to be equal to init_radio_delay - ConfigEndDuration before to start the final configuration. The time unit for this delay is 1us.
15:8	TXDATAREADYCHECK	0x0	RW	Duration for the BLE sequencer to get the TxDataReady and DatPtr information in TxRxPack table. The goal of this bit field is to provide more time to the Firmware to provide the data pointer address and in case of transmission to provide the data to transmit. The BLE sequencer waits for relative time to be equal to init_radio_delay - TxdataReadyCheck before to start the final configuration. The time unit for this delay is 1us.
23:16	TXDELAYSTART	0x0	RW	Delay added between the moment the Radio FSM is in TX mode (PA ramp up done and power present on the antenna) and the first bit transmission to the modulator. The time unit for this delay is 125ns.
29:24	TXDELAYEND	0x0	RW	Delay added between the last bit transmission to the modulator and the "end of transmission" information for the BLE sequencer. The time unit for this delay is 125ns. This delay allows giving time to the modulator and analog chain to output on the antenna the last bit.
30	TIMECAPTURESEL	0x0	RW	0: the captured time (absolute time) corresponds to the end of 1st INIT step in the BLE sequence (InitDelay timeout event). 1: the captured time (absolute time) corresponds to the end of DATA INIT step in the BLE sequence (init_radio_delay timeout event). Note: This bit is for debug purpose.
31	TIMECAPTURE	0x0	RW	0: no capture is requested to monitor the BLE sequence. 1: a time capture is requested to monitor the BLE sequence. Captured event is defined by GlobalStatMach.TIMECAPTURESEL bit. Note: If both TIMECAPTURE and TIMECAPTURESEL bits are low, the TimerCaptureReg BLE APB register is anyway updated with the InitDelay timeout event (mechanism to bypass the fact those 2 GlobalStatMach bits are checked after 1st INIT step completion). Note: If TxRxPack.TrigRcv or TxRxPack.TrigDone bit is set, the TimerCaptureReg BLE APB register shows this last event trig value at the end. Note: This bit is for debug purpose.

Table 128. **GlobalStatMach.WORD4 register description**

Bit	Field name	Reset	RW	Description
7:0	TXREADYTIMEOUT	0x0	RW	Transmission ready timeout. Defines the maximum duration for the transmit block to wait for the Radio FSM to indicate it is in TX state and data can be provided to the modulator. The time unit for this delay is 1us. Note: If this value is set to 0, no timeout is activate to wait the TX ready information.
27:8	RCVTIMEOUT	0x0	RW	Receive window timeout. Define the maximum duration to stay in reception without any preamble + access address detection (rest of the frame can be received even outside this time window). The duration is expressed as (4^RCVTIMEOUT[19;18]) x RCVTIMEOUT[17:0] The time unit for RCVTIMEOUT[17:0] is 1us.
31:28	RESERVED31_28	0x0	RW	Ignored on write - read as zero

Table 129. GlobalStatMach.WORD5 register description

Bit	Field name	Reset	RW	Description
0	AUTOTXRXSKIPEN	0x0	RW	Automatic transfer (TX or RX) skip enable. If set, the BLE link layer stops automatically an on-going transfer if PLL lock fail event is detected on PLL start.
1	RESERVED1	0x0	RW	Ignored on write - read as zero
2	CHKFLAGAUTOCLEARENA	0x0	RW	Active Auto Clear bit Enable. The Active auto clear feature leads the sequencer to clear the GlobalStatMach.Active bit during the RAM write back step at the end of a transfer/sequence. The main goal of this feature is to avoid a new transfer to start on the antenna while the software did not yet prepare the next transfer in RAM tables. 0: the active auto clear bit feature is disabled. 1: The active auto clear bit feature is enabled.
5:7	RESERVED	0x0	RW	Reserved.
12:8	INTSEQERROR	0x0	RW	Sequencer errors interrupt enable. For each bit of IntSeqError[4:0], the associated SeqError[x] flag (located in DebugStatusReg APB BLE register) generates an interrupt on the int3 line (debug interrupt).
19:13	RESERVED19_13	0x0	RW	Ignored on write - read as zero
20	INTADDPOINTERROR	0x0	RW	Address pointer error interrupt enable. 0: the interrupt associated to Interrupt1Reg.AddPointError is disabled. 1: the interrupt associated to Interrupt1Reg.AddPointError is enabled.
21	INTALLTABLEREADYERROR	0x0	RW	All table ready error interrupt enable. 0: the interrupt associated to Interrupt1Reg.AllTableReadyError is disabled. 1: the interrupt associated to Interrupt1Reg.AllTableReadyError is enabled.
22	INTTXDATAREADYERROR	0x0	RW	Transmission data payload ready error interrupt enable. 0: the interrupt associated to Interrupt1Reg.TxDataReady is disabled. 1: the interrupt associated to Interrupt1Reg.TxDataReady is enabled.
23	INTNOACTIVELERROR	0x0	RW	Active bit low value reading interrupt enable. 0: the interrupt associated to Interrupt1Reg.NoActiveLError is disabled. 1: the interrupt associated to Interrupt1Reg.NoActiveLError is enabled.
24	RESERVED	0x0	RW	Reserved
25	INTRCVLENGTHERROR	0x0	RW	Too long received payload length interrupt enable. 0: the interrupt associated to Interrupt1Reg.ReceiveLengthError is disabled. 1: the interrupt associated to Interrupt1Reg.ReceiveLengthError is enabled.
26	INTSEMATIMEOUTERROR	0x0	RW	Semaphore timeout error interrupt enable. 0: the interrupt associated to Interrupt1Reg.SemaTimeoutError is disabled. 1: the interrupt associated to Interrupt1Reg.SemaTimeoutError is enabled.
27	RESERVED	0x0	RW	Reserved.
28	INTSEQDONE	0x0	RW	Sequencer end of task interrupt enable. This bit should always be set to ensure an interrupt occurs at the end of sequence whatever the exit reason. 0: the interrupt associated to Interrupt1Reg.SeqDone is disabled. 1: the interrupt associated to Interrupt1Reg.SeqDone is enabled.
29	INTTXRXSKIP	0x0	RW	Transmission or reception skip interrupt enable. 0: the interrupt associated to Interrupt1Reg.intTxRxSkip is disabled. 1: the interrupt associated to Interrupt1Reg.intTxRxSkip is enabled.
30	INTACTIVE2ERR	0x0	RW	no initialization_finished from Radio FSM received on time interrupt enable.

Bit	Field name	Reset	RW	Description
				0: the interrupt associated to Interrupt1Reg.Active2Error is disabled. 1: the interrupt associated to Interrupt1Reg.Active2Error is enabled.
31	INTCONFIGERROR	0x0	RW	Configuration error interrupt enable. 0: the interrupt associated to Interrupt1Reg.ConfigError is disabled 1: the interrupt associated to Interrupt1Reg. ConfigError is enabled.

Table 130. GlobalStatMach.WORD6 register description

Bit	Field name	Reset	RW	Description
31:0	RESERVED31_0	0x0	RW	Ignored on write - read as zero

6.2 StatMach RAM table

The StatMach table links to an active connection. There are as many StatMach tables as concurrent connections in a limit of 128 (maximum supported by the hardware).

The StatMach RAM table locations are frozen by the hardware as they follow the GlobalStatMach. The formula for a StatMach base address is:

StateMachBaseAddress[stateMachIdx] = GlobStatMachBaseAddress + 28 + (stateMachIdx * 80)

Table 131. StatMach

Address offset	Name	RW	Reset	Description
0x00	WORD0	RW	0x00000000	Word0 register
0x04	WORD1	RW	0x00000000	Word1 register
0x08	WORD2	RW	0x00000000	Word2 register
0x0C	WORD3	RW	0x00000000	Word3 register
0x10	WORD4	RW	0x00000000	Word4 register
0x14	WORD5	RW	0x00000000	Word5 register
0x18	WORD6	RW	0x00000000	Word6 register
0x1C	WORD7	RW	0x00000000	Word7 register
0x20	WORD8	RW	0x00000000	Word8 register
0x24	WORD9	RW	0x00000000	Word9 register
0x28	WORDA	RW	0x00000000	WordA register
0x2C	WORDB	RW	0x00000000	WordB register
0x30	WORDC	RW	0x00000000	WordC register
0x34	WORDD	RW	0x00000000	WordD register
0x38	WORDE	RW	0x00000000	WordE register
0x3C	WORDF	RW	0x00000000	WordF register
0x40	WORD10	RW	0x00000000	Word10 register
0x44	WORD11	RW	0x00000000	Word11 register
0x48	WORD12	RW	0x00000000	Word12 register

Table 132. StatMach.WORD0 register description

Bit	Field name	Reset	RW	Description
5:0	UCHAN	0x0	RW	BTLE unmapped channel index. UChan is used by the channel incrementer and the remapper to generate a new Uchan and RemapChan values through the two algorithms defined by the Bluetooth core 5.0 specification. Note: This field is written back at the end of the transfer by the sequencer: - if TxRxPack.incchan = 0, written back value is the same value, - if TxRxPack.incchan = 1, written back value is the value modified by one of the two algorithms defined by the Bluetooth core 5.0 specification. Note: The standard requests this bit field to be set to 0 for the first connection event.
6	RESERVED	0x0	RW	Reserved.
7	TXMODE	0x0	RW	Transfer type selection of the current sequence. This bit is re-written by the sequencer with StatMach.NextTxMode bit value during each RAM write back phase. 0: requested transfer is a reception. The start address of the TxRxPack packet in which the received data has to be stored is pointed by rcvpoint. 1: requested transfer is a transmission. The start address of the TxRxPack packet to be transmitted is pointed by TxPoint.
13:8	REMAP_CHAN	0x0	RW	BTLE Remapped channel index. This is the remapped channel as described in algorithm1 and algorithm2 in BlueNRG core specification 5.0. This bit field is used by the hardware to generate the physical channel frequency. Note: This field is written back at the end of the transfer by the sequencer: - if TxRxPack.incchan = 0, written back value is the same value, - if TxRxPack.incchan = 1, written back value is the value modified by one of the two algorithms defined by the Bluetooth core 5.0 specification and mapped to the used channels list. Note: The standard requests this bit field to be set to 0 for the first connection event.
14	SN	0x0	RW	BTLE sequence number bit. If TxRxPack.SN_EN = 0 or TxRxPack.Advertise = 1, this bit is kept unchanged at the end of a transfer. If TxRxPack.SN_EN = 1 and TxRxPack.Advertise = 0, this bit is managed automatically by the hardware SN/NESN mechanism (as described in the BlueNRG core specification 5.0). Then, this bit is modified by the hardware only at the end of a reception (not on transmission). Note: In any case, this bit is written back by the sequencer at the end of a transfer (modified or not).
15	NESN	0x0	RW	BTLE next expected sequence number bit. If TxRxPack.SN_EN=0 or TxRxPack.Advertise=1, this bit is kept unchanged at the end of a transfer. If TxRxPack.SN_EN = 1 and TxRxPack.Advertise = 0, this bit is managed automatically by the hardware SN/NESN mechanism (as described in the BlueNRG core specification 5.0). Then, this bit is modified by the hardware only at the end of a reception (not on transmission). Note: In any case, this bit is written back by the sequencer at the end of a transfer (modified or not)
18:16	RESERVED	0x0	RW	Reserved.
19	RESERVED	0x0	RW	Reserved.
20	BUFFER_FULL	0x0	RW	No more receive buffer available. Set this bit to indicate no more buffer is available to receive any packet. In this case: - no data are written back in the RAM at the end of the sequence - the SN/NESN automatic mechanism adapts its behavior by keeping the NESN unchanged and does not increment the encryption receive packet counter. Note: The SN bit management is not impacted to keep the transmission progressing as long as the peer acknowledges the reception of previous transmitted packet.

Bit	Field name	Reset	RW	Description
21	ENCRYPTON	0x0	RW	"On the fly" encryption/decryption engine enable. 0: the "On the fly" encryption/decryption engine is disabled. 1: the "On the fly" encryption/decryption engine is enabled. The parameters StateMach.EncryptIV and StateMach.EncryptK are read from RAM during the initialization phase. Note: The "On the fly" encryption/decryption engine does not run for packet with null length. Note: It is mandatory to have TxRxPack.SN_EN = 1 when StateMach.Encryption = 1 as PCntTx is incremented by the SN/NESM automatic management mechanism.
22	TXENC	0x0	RW	Previous transmission packet was encrypted. Note: This bit is fully managed by the hardware. It is set to 1 after the transmission of an encrypted packet (so with length not zero). When TxEnc = 0, PCntTx (transmission packet counter required for the sub-keys calculation) is unchanged. When TxEnc = 1, PCntTx may be incremented depending on the SN/NESN check result.
23	RCVENC	0x0	RW	Last rreceive packet was encrypted. Note: This bit is fully managed by the hardware. It is set to 1 after the reception of a packet with length not zero (whatever the CRC check result) if StateMach.Encryption = 1. When RcvEnc = 1, the PCntRcv (receive packet counter required for the sub-keys calculation) is incremented depending on the SN/NESN check result.
26:24	TXPHY	0x0	RW	Transmission Phy selection. -000: selected transmitter PHY is legacy 1 Mbps -001: selected transmitter PHY is legacy 2 Mbps -100: selected transmitter PHY is coded 1 Mbps with S=8 -110: selected transmitter PHY is coded 1 Mbps with S=2 others: reserved for future use. If programmed by mistake, selects "Transmitter PHY is not coded 1 Mbps" option.
27	RESERVED27	0x0	RW	Ignored on write -read as zero
30:28	RXPHY	0x0	RW	Reception Phy selection. bit0: bit rate (0=1 Mbps / 1=2 MBps) / bit1: does not care / bit2: coded/not coded. - 000: selected receiver PHY is legacy 1 Mbps - 001: selected receiver PHY is legacy 2 Mbps - 1x0: selected receiver PHY is coded 1 Mbps - others: reserved for future use. If programmed by mistake, selects "Receiver PHY is not coded 1 Mbps" option. Note: S2/S8 coded choice comes from an auto-detection done by the demodulator.
31	RESERVED31	0x0	RW	Ignored on write - read as zero

Table 133. StatMach.WORD1 register description

Bit	Field name	Reset	RW	Description
31:0	TXPOINT	0x0	RW	Pointer to transmit packet. TxPoint defines the start address of the TxRxPack link list (containing the parameters of the current transmission to be proceeded). This variable needs to be initialized by the firmware with the start address of the first TxRxPack of the transmission linked list each time a StateMach is created in memory (new connection). Then, TxPoint is managed by the hardware, considering the firmware has to guarantee the transmission link list is never empty (or pointing to itself). Note: This pointer address must be 32-bit aligned and is an absolute address (not an offset).

Table 134. **StatMach.WORD2 register description**

Bit	Field name	Reset	RW	Description
31:0	RCVPOINT	0x0	RW	Pointer to receive packet. Rcvpoint defines the start address of the TxRxPack link list (containing the parameters of the current reception to be proceeded) This variable needs to be initialized by the firmware with the start address of the first TxRxPack of the reception linked list each time a StateMach is created in memory (new connection). Then, RcvPoint is managed by the hardware, considering the firmware has to guarantee the reception link list is never empty (or pointing to itself). Note: This pointer address must be 32-bit aligned and is an absolute address (not an offset).

Table 135. **StatMach.WORD3 register description**

Bit	Field name	Reset	RW	Description
31:0	TXPOINTPREV	0x0	RW	Pointer to previous transmit packet. This variable is fully managed by the hardware. It is recommended to initialize to 0 by the firmware when the StateMach is created in memory (new connection). TxPointPrev indicates which buffer can be reallocated (as it is now free).

Table 136. **StatMach.WORD4 register description**

Bit	Field name	Reset	RW	Description
31:0	RCVPOINTPREV	0x0	RW	Pointer to previous receive packet. This variable is fully managed by the hardware. It is recommended to initialize to 0 by the firmware when the StateMach is created in memory (new connection). RcvPointPrev indicates which buffer can be reallocated (as it is now free).

Table 137. **StatMach.WORD5 register description**

Bit	Field name	Reset	RW	Description
31:0	TXPOINTNEXT	0x0	RW	Next transmit pointer. This variable is fully managed by the hardware. It is recommended to initialize to 0 by the firmware when the StateMach is created in memory (new connection). TxPointNext indicates the address of the TxRxPack transmit packet to be used once the transmission managed by the TxPoint is done (TxRxPack.NextPtr[31:0]). The TxPointNext bit field is always updated at the end of a transmission. Note: At the end of a valid reception with TxRxPack.SN_EN = 1 and TxRxPack.Advertise = 0, the StatMach.TxPoint is equal to the StatMach.TxPointNext.

Table 138. **StatMach.WORD6 register description**

Bit	Field name	Reset	RW	Description
31:0	PCNTTX_31_0	0x0	RW	CCM encryption transmission packet counter [31:0]. PCntTx is used during the on the fly encryption of the transmission data by the AES encryption engine. For each new connection, Bluetooth protocol requires PCntTx to be initialized by the firmware to the value: - 40'h8000000000: for Data Channel PDUs sent by the master - 40'h0000000000: for Data Channel PDUs sent by the slave. Note: It is mandatory to have TxRxPack.SN_EN = 1 when StateMach.Encryption = 1 as PCntTx is incremented by the SN/NESM automatic management mechanism.

Table 139. StatMach.WORD7 register description

Bit	Field name	Reset	RW	Description
7:0	PCNTTX_39_32	0x0	RW	CCM encryption transmission packet counter [39:32]. PCntTx is used during the on the fly encryption of the transmission data by the AES encryption engine. For each new connection, Bluetooth protocol requires PCntTx to be initialized by the firmware to the value: - 40'h8000000000: for Data Channel PDUs sent by the master - 40'h0000000000: for Data Channel PDUs sent by the slave. Note: It is mandatory to have TxRxPack.SN_EN = 1 when StateMach.Encryption = 1 as PCntTx is incremented by the SN/NESM automatic management mechanism.
31:8	PCNTRCV_23_0	0x0	RW	CCM encryption Receive Packet counter [23:0]. PCntRcv is used during the on the fly encryption of the received data by the AES encryption engine. For each new connection, Bluetooth protocol requires PCntRcv to be initialized by the firmware to the value: - 40'h8000000000: for Data Channel PDUs received by the slave - 40'h0000000000: for Data Channel PDUs received by the master. Note: It is mandatory to have TxRxPack.SN_EN = 1 as PCntRcv is incremented by the SN/NESM automatic management mechanism.

Table 140. StatMach.WORD8 register description

Bit	Field name	Reset	RW	Description
15:0	PCNTRCV_39_24	0x0	RW	CCM encryption Receive Packet counter [39:24]. PCntRcv is used during the on the fly encryption of the received data by the AES encryption engine. For each new connection, Bluetooth protocol requires PCntRcv to be initialized by the firmware to the value: - 40'h8000000000: for Data Channel PDUs received by the slave - 40'h0000000000: for Data Channel PDUs received by the master. Note: It is mandatory to have TxRxPack.SN_EN = 1 as PCntRcv is incremented by the SN/NESM automatic management mechanism.
19:16	PREAMBLEREP	0x0	RW	Transmission Preamble Repetition number. Defines the number of repetition of the transmitted preamble length for coded or uncoded phy. Keep it at 0 to have the normal preamble format (1 byte). Note: If StateMach.EnaPreambleRep = 0, this bit field is not taken into account. This feature is not Bluetooth standard.
20	ENAPREAMBLEREP	0x0	RW	Enable transmission preamble repetition. 0: the preamble feature is disabled and the preamble length is as described in the core specification 5.0. 1: The preamble feature is enabled and the preamble length is defined by StateMach.PreambleRep (for coded and uncoded phy). This feature is not Bluetooth standard.
21	DISABLECRC	0x0	RW	CRC Disable. If set, this bit: - in reception: disable the check of the CRC - in transmission: no CRC field is generated nor inserted in the sent packet. This feature is not Bluetooth standard.
22	RESERVED	0x0	RW	Reserved.
23	RXMICDBG	0x0	RW	Receive MIC debug

Bit	Field name	Reset	RW	Description
				0: the decrypted MIC (locally computed) is stored in the payload buffer in RAM (at the end of the payload). 1: the received MIC is stored in the payload buffer in RAM (at the end of the payload). This feature is for debug.
28:24	INTTXERROR	0x0	RW	Transmission error interrupt enable. If IntTxError[n] = 1: an interrupt is generated and associated flag is set in Interrupt1Reg.TxError[n] if a TxError[n] event occurs during the transmission. If IntTxError[n] = 0: no interrupt nor associated flag in Interrupt1Reg.TxError[n] is available if a TxError[n] event occurs during the transmission. Note: StatusReg.TxError[n] bit is not impacted and always provides the TxError[n] unmasked information.
29	INTENCERROR	0x0	RW	Receive encryption error interrupt enable. 0: the receive encryption error interrupt is disabled. 1: the receive encryption error interrupt is enabled (and associated interrupt flag is visible in Interrupt1Reg.EncError. The interrupt is active if the MIC of the received message does not match the computed one (while the preamble and the access address are received ok, StateMach.Encryption = 1 and the received length is not null). Note: The CRC check result is not taken into account for this interrupt.
30	INTRXOVERFLOWERROR	0x0	RW	Receive data path overflow error interrupt enable 0: the interrupt Interrupt1Reg.IntRxOverflowError is disabled 1: the interrupt Interrupt1Reg.IntRxOverflowError is enabled.
31	RXDEBUGCRC	0x0	RW	Debug mode of the CRC in reception 0: the received CRC is not saved with payload in RAM (this is the normal mode) 1: the received CRC is saved with payload in RAM (this is a debug mode). When set: • the packet is accepted whatever the CRC: so if CRC error, then the RCVOK flag is set anyway and no CRC error flag is raised • the DataPack RAM buffer size must take into account the 3 additional CRC bytes

Table 141. StatMach.WORD9 register description

Bit	Field name	Reset	RW	Description
31:0	ACCADDR	0x0	RW	BlueNRG packet access address. This value is used in transmission and in reception. - in transmission, it is inserted in the packet after the preamble. - in reception, it is used by the demodulator to detect and accept a received packet. Note: The nature of a packet (primary advertising, secondary advertising or data) is only defined by TxRxPack.Advertise so StateMach.Accadr = 0x8E89BED6 does not mean that the packet is an advertising packet.

Table 142. StatMach.WORDA register description

Bit	Field name	Reset	RW	Description
23:0	CRCINIT	0x0	RW	CRC initialization value. This value is used to initialize the CRC for Data packet or for AUX_SYNC_IND PDU and its subordinate set. This field is ignored if TxRxPack.CRCINITSEL = 0.
31:24	MAXRECEIVEDLENGTH	0x0	RW	Maximum receive length. Defines the maximum receive length the BLE link controller can accept.

Bit	Field name	Reset	RW	Description
				If the length of the received packet is greater than this value, the hardware limits the payload RAM write back data to the defined maximum length and stops the reception treatment on this defined maximum length (implying also CRC error, etc.) The ReceiveLengthError event is raised (visible in StatusReg and if associated interrupt is enabled in Interrupt1Reg register). The received packet is processed normally when the received length located in the received packet header is smaller or equal to StateMach.MaxReceivedLength.

Table 143. **StatMach.WORDB register description**

Bit	Field name	Reset	RW	Description
4:0	PAPOWER	0x0	RW	32 power levels are available. From 0 to 0x1F.
7:5	RESERVED7_5	0x0	RW	Ignored on write - read as zero
13:8	HOPINCR	0x0	RW	Hop increment. Defines the hop increment as described in the algorithm 1 of the BlueNRG 5.0 core specification.
15:14	RESERVED15_14	0x0	RW	Ignored on write - read as zero
31:16	USEDCHANNELFLAGS_15_0	0x0	RW	Remapping flags[15:0] for all 37 BTLE channels. The remapping flags are used by the Bluetooth smart algorithm 1 and 2. If bit(n]) = 1, the channel n may be used for reception or transmission. If bit(n) = 0, the channel n cannot be used for reception or transmission. Note: This parameter is described in channel classification/ channel map in the Bluetooth core specification 5.0.

Table 144. **StatMach.WORDC register description**

Bit	Field name	Reset	RW	Description
21:0	USEDCHANNELFLAGS_36_16	0x0	RW	Remapping flags[3616] for all 37 BTLE channels. The remapping flags are used by the Bluetooth smart algorithm 1 and 2. If bit(n]) = 1, the channel n may be used for reception or transmission. If bit(n) = 0, the channel n cannot be used for reception or transmission. Note: This parameter is described in channel classification/ channel map in the Bluetooth core specification 5.0.
31:22	RESERVED31_22	0x0	RW	Ignored on write - read as zero

Table 145. **StatMach.WORDD register description**

Bit	Field name	Reset	RW	Description
15:0	CONNEVENTCOUNTER	0x0	RW	Connection event counter value. Contains a copy of the connection event counter value, used by the channel incrementer to compute the algorithm #2. This bit field has to be managed by the SW.
31:16	PAEVENTCOUNTER	0x0	RW	Advertising event counter value. Contains a copy of the Advertising event counter value, used by the channel incrementer to compute the algorithm #2. This bit field has to be managed by the SW.

Table 146. **StatMach.WORDE register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTIV_31_0	0x0	RW	Initial vector for encryption [31:0]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

Table 147. **StatMach.WORDF register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTIV_63_32	0x0	RW	Initial vector for encryption [63:32]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

Table 148. **StatMach.WORD10 register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTK_31_0	0x0	RW	Encryption key [31:0]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

Table 149. **StatMach.WORD11 register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTK_63_32	0x0	RW	Encryption key [63:32]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

Table 150. **StatMach.WORD12 register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTK_95_64	0x0	RW	Encryption key [95:64]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

Table 151. **StatMach.WORD13 register description**

Bit	Field name	Reset	RW	Description
31:0	ENCRYPTK_127_96	0x0	RW	Encryption key [127:96]. This value is used by the AES engine during on the fly AES CCM encryption. See Bluetooth Low Energy CCM encryption description in BTLE core spec 5.0.

6.2.1 PaPower bit field description

The table below provides the PA power correspondence to program the StateMach.PaPower bit field.

Table 152. StatMach.PaPower values

Value (Hexa)	Output power (dBm)	Value (Hexa)	Output power (dBm)	Value (Hexa)	Output power (dBm)	Value (Hexa)	Output power (dBm)
1F	+6/+8[(1)]	17	-0.5	F	-5.9	7	-14.1
1E	+5	16	-0.85	E	-6.9	6	-15.25
1D	+4	15	-1.3	D	-7.8	5	-16.5
1C	+3	14	-1.8	C	-8.85	4	-17.6
1B	+2	13	-2.45	B	-9.9	3	-18.85
1A	+1	12	-3.15	A	-10.9	2	-19.75
19	0	11	-4	9	-12.05	1	-20.85
18	-0.15	10	-4.95	8	-13.15	0	-40

1. *Several settings are needed to reach the +8 dBm in transmission:*
   - *Program the SMPS located in the SoC to provide 1.85 V*
   - *Program 0x1F in StatMach.PaPower[4:0] bit field*
   - *Configure the LDO_TRANSFO in bypass mode by setting radio register LDO_ANA_ENG[1] = RFD_LDO_TRANSFO_BYPASS = 1*

   Warning: the LDO_ANA_ENG[1] = RFD_LDO_TRANSFO_BYPASS bit must be reset in reception.

6.3 TxRxPack RAM table

The firmware has to guarantee that the transmission/reception link list is never empty or at least it must point to itself.

Table 153. TxRxPack

Address offset	Name	RW	Reset	Description
0x00	WORD0	RW	0x00000000	Word0 register
0x04	WORD1	RW	0x00000000	Word1 register
0x08	WORD2	RW	0x00000000	Word2 register
0x0C	WORD3	RW	0x00000000	Word3 register
0x10	WORD4	RW	0x00000000	Word4 register

Table 154. TxRxPack.WORD0 register description

Bit	Field name	Reset	RW	Description
31:0	NEXTPTR	0x0	RW	Next pointer address entry of the linked list. Points to the next transmit or receive packet. The user must enter the absolute address, not an offset. Caution: This pointer must be 32-bit aligned or else StatusReg.AddPointError is set (and Interrupt1Reg.AddPointError if GlobalStatMach.IntAddPointError = 1).

Table 155. TxRxPack.WORD1 register description

Bit	Field name	Reset	RW	Description
0	CALREQ	0x0	RW	Calibration request.

Bit	Field name	Reset	RW	Description
				0: the radio frequency and KVCO2 calibration is disabled. This setting is used when this calibration has already been done and if the radio did not go to low power state. 1: The calibration of the radio frequency, KVCO2, the complex pass band filter and the PLL is enabled. It must be performed at each channel frequency change or after the wakeup.
1	CHANALGO2SEL	0x0	RW	Channel hoping algorithm selection. if TxRxPack.incchan = 0, this bit field has no effect. if TxRxPack.incchan = 1: 0: the algorithm #1 is used for the channel hoping for data channel. For primary advertising, channels are automatically incremented as follows: 37->38->39->37->etc. 1: The algorithm #2 is used for the channel hoping in data connection or for periodic advertising packets.
2	KEEPSEMAREQ	0x0	RW	Request to keep the radio token active at the end of the current transfer.. Caution: This bit MUST be set to fit the IFS = 150 us constraint. Indeed, when the token is released, the Radio FSM switches back to IDLE mode. The radio FSM needs around 60 us more (VBG_BOOST and ENA_CUR states) to go back to ACTIVE2 state on next BLE trig event.
3	RESERVED	0x0	RW	Reserved. It must be kept at 0.
4	CRCINITSEL	0x0	RW	CRC initialization value selector. 0: the transmit and the receive block initialize their CRC with a constant equal to: 0x555555 1: the transmit and the receive block initialize their CRC with the value defined by StateMach.CrcInit
5	ADVERTISE	0x0	RW	Advertise packet format 0: the packet format stored in RAM or to be received is a data packet format. 1: The packet format stored in RAM or to be received is an advertise packet format.
6	SN_EN	0x0	RW	Automatic SN, NESN hardware mechanism enable. 0: automatic SN/NESN hardware mechanism is disabled. The receive pointers and transmit pointers are systematically shifted independently of SN, NESN bits and also on a receive timeout sequence. 1: Automatic SN/NESN hardware mechanism is enabled.
7	INCCHAN	0x0	RW	Automatic channel incrementer enable. When enabled, the automatic channel incrementer takes as input StateMach.UChan, TxRxPack.Advertise, TxRxPack.ChanAlgo2Sel, StateMach.Remap_chan, StateMach.hopincr, StateMach.UsedChannelFlags, StateMach.connEventCounter and StateMach.paEventCounter. 0: automatic channel incrementer is disabled 1: automatic channel increment is enabled. See Channel number management section for details.
8	NEXTTXMODE	0x0	RW	Flag indicating if next TxRx packet to be handled by the link controller StateMach is a receive packet or a transmit packet. The BLE sequencer overloads StateMach.TxMode value with NextTxMode value during each RAM write back phase. 0: next TxRx packet is a receive packet. 1: next TxRx packet is a transmit packet.
9	ALLTABLEREADY	0x0	RW	All table data ready. This bit is checked at the beginning of the 2^{nd} INIT phase to ensure bit fields related to on-going transfer and about to be read are relevant. 0: the RAM table information related to the on-going transfer are not ready. The transmission is not started by the sequencer. 1: The RAM table information related to the on-going transfer are ready. The transmission is started by the sequencer. Note: The goal of this bit is to allow the software blocking a transfer if RAM table update is not over.
10	TXDATAREADY	0x0	RW	Transmission data ready. This bit is checked only if the current transfer is a transmission. The check is done at the beginning of the DATA INIT phase to ensure the at least a few bytes of the transmission payload are already written in the data buffer.

Bit	Field name	Reset	RW	Description
				This bit allows doing an "On-the-fly" data buffer memcopy while transmission has already started on the antenna. 0: the transmission payload is not ready. The transfer is not started by the sequencer. 1: The transmission payload is ready so the transfer is started by the sequencer. Note: The recommendation for transmission data payload is to set this TxDataReady bit only when at least 16 bytes of data are available in the payload data buffer.
11	RESERVED	0x0	RW	Reserved. It must be kept at 0.
12	DISABLEWHITENING	0x0	RW	Whitening Disable 0: the whitening is enabled in the transmit block and in the receive block. 1: The whitening is disabled in the transmit block and in the receive block. This may be used for debug or during official Bluetooth compliance test.
31:13	RESERVED31_13	0x0	RW	Reserved.

Table 156. **TxRxPack.WORD2 register description**

Bit	Field name	Reset	RW	Description
31:0	DATAPTR	0x0	RW	Data pointer address. Points to the data packet linked with TxRxPack (called DataPack in this document). This data packet contains the header and the data, excluding the preamble, the access address and the CRC. The BLE link layer writes this packet in RAM in case of reception and reads it from RAM in case of transmission. Note: This pointer has no memory address alignment requirement. However the software must write an absolute address (not an offset). If the 8-bit MSB part of the pointer value is not equal to the RAM 8-bit MAB address, an AddPointError flag is raised.

Table 157. **TxRxPack.WORD3 register description**

Bit	Field name	Reset	RW	Description
19:0	TIMER2	0x0	RW	Timer2 triggering value setting. Defines the delay before next Timer2 trigger event if TxRxPack.Timer2En = 1. Time unit is in microseconds. Note: The Timer2 delay starts at the end of the on-going sequence.
20	TIMER2EN	0x0	RW	Timer2 enable (for next timer trig). 0: Timer2 disabled at the end of this current packet. 1: Timer2 is enabled at the end of this current packet.
21	RESERVED21	0x0	RW	Ignored on write - read as zero
22	TRIGRCV	0x0	RW	Time capture enable on received preamble and access address pattern detection. 0: no time stamping requested on preamble + access address detection. 1: The interpolated absolute time is captured in CurrentTimeReg.TimerCaptureReg when the demodulator detects the preamble + access address in the received bit stream. When this bit is set and if a time capture occurs, the StatusReg.TimeCaptureTrig is set to 1. An interrupt is raised if enabled (associated to Interrupt1Reg. TimeCaptureTrig set to 1). This bit must be set to 0 in transmission TxRxPack table not to disturb other time capture options. Note: If GlobalStatMach.TimeCapture or TxRxPack.TrigDone bit is set, the TimerCaptureReg BLE APB register shows this last event trig value at the end.
23	TRIGDONE	0x0	RW	Time capture enable on "On air" last transmitted/received bit.

Bit	Field name	Reset	RW	Description
				0: no time stamping in CurrentTimeReg.TimerCaptureReg is achieved, no interrupt is generated by TrigDone. 1: The interpolated absolute time is captured in CurrentTimeReg.TimerCaptureReg when the demodulator receives the last bit of the bit stream or when the last transmitted has been shifted out of the transmit block. When this bit is set and if a time capture event occurs, the StatusReg.TrigDone is set to 1. An interrupt is raised if enabled (associated to Interrupt1Reg.TrigDone set to 1). Note: If GlobalStatMach.TimeCapture or TxRxPack.TrigRcv bit is set, the TimerCaptureReg BLE APB register shows this last event trig value at the end.
24	INTTXOK	0x0	RW	Interrupt enable of "good reception of transmitted packet is confirmed by the peer device". 0: the interrupt Interrupt1Reg.TxOk is Disabled 1: The interrupt Interrupt1Reg.TxOk is enabled Note: This interrupt has to be enabled in the RxPack table as the feature is active at the end of a reception.
25	INTDONE	0x0	RW	Done interrupt enable 0: the interrupt Interrupt1Reg.Done is Disabled 1: The interrupt Interrupt1Reg.Done is enabled
26	INTRCVTIMEOUT	0x0	RW	Receive timeout interrupt enable 0: the interrupt Interrupt1Reg.RcvTimeout is Disabled 1: The interrupt Interrupt1Reg.RcvTimeout is enabled
27	INTRCVNOMD	0x0	RW	No more Data (end of connection found) interrupt enable 0: the interrupt Interrupt1Reg.RcvNoMd is Disabled 1: The interrupt Interrupt1Reg.RcvNoMd is enabled
28	INTRCVCMD	0x0	RW	"Received packet is a command" interrupt enable 0: the interrupt Interrupt1Reg.RcvCmd is Disabled 1: The interrupt Interrupt1Reg.RcvCmd is enabled
29	INTTIMECAPTURE	0x0	RW	"Time Capture occurred" interrupt enable 0: the interrupt Interrupt1Reg.IntTimeCaptureTrig is Disabled 1: The interrupt Interrupt1Reg. IntTimeCaptureTrig is enabled Note: The event(s) responsible for the interrupt can be the sequencer Time Capture and/or the TrigDone and/or the TrigRcv events.
30	INTRCVCRCERR	0x0	RW	Receive CRC error interrupt enable 0: the interrupt Interrupt1Reg.RcvCrcerr is Disabled 1: The interrupt Interrupt1Reg.RcvCrcerr is enabled
31	INTRCVOK	0x0	RW	Receive OK interrupt enable 0: the interrupt Interrupt1Reg.RcvOk is Disabled 1: The interrupt Interrupt1Reg.RcvOk is enabled

Table 158. TxRxPack.WORD4 register description

Bit	Field name	Reset	RW	Description
31:0	RESERVED	0x0	RW	Reserved.

A similar feature allows using the existing slow clock timer to generate a CPU wake-up source. This feature when activated has no impact on the BLE transfers (no trigger event generated to the BLE sequencer).

In this case, the CPU wake-up process occurs in two steps:

- ABSOLUTE_TIME[31:4] = CM0_WAKEUP_TIME [31:4] - Wakeup.WAKEUP_OFFSET[7:0]
- The wake-up block raises an SoC wake-up request towards the Power Controller of the SoC to request voltage/clock restoring
- ABSOLUTE_TIME[31:4] = CM0_WAKEUP_TIME [31:4]
- If WAKEUP_CM0_IRQ_ENABLE.WAKEUP_IT = 1, the wake-up block sends an interrupt towards the CPU (irq_wakeup_cpu line)

7 Wakeup block

The wake-up block is partially located in the always-on power domain to stay supplied even in the low power modes of the device. All features not mandatory during low power modes are located in the 1V2 switchable power domain to limit power consumption.

The wakeup block combines in fact two features:

- wakeup / sleep requests management
- absolute and interpolated time computation.

The wake-up block computes two kinds of time: the absolute time and the interpolated time.

7.1 Absolute time

This timer is located in the always-on power domain and is based on a rollover free running counter. The absolute time is computed by a 28-bit counter clocked on the slow clock (around 32 kHz).

This absolute time:

- is used by the BLE controller and by the CPU to trigger a wakeup. When the current absolute time reaches the wake-up time programmed, then a wake-up signal is generated either for the controller or for the CPU
- is provided to the time interpolator block to build the 28-bit MSB non-interpolated part of the 32-bit interpolated time.

7.2 Interpolated time

The interpolated time is located in the 1.2 V switchable power domain and is clocked at 16 x slow clock frequency (generated from the system clock). This interpolated time is a 32-bit timer built with:

- 28-bit MSB part corresponding to the non-interpolated time clocked at 32 kHz (absolute time)
- 4-bit LSB corresponding to the fractional part (interpolation at 512 kHz).

The interpolated time is provided to the BLE controller to get current time information and to manage the timer1.

7.3 Sleep request and wakeup management

The wake-up block offers the interface to issue the sleep request to the power controller from BLE controller and it raises wake-up requests for the SoC and the BLE controller.

The sleep request, coming from BLE controller, can be performed through a wake-up block register.

If the wakeup is enabled, two separate IRQs are asserted in order to:

- indicate to the CPU that the BLE controller receives a wake-up request from the wake-up block. The wake-up request source is the wake-up timer.
- indicate that the CPU wake-up timer reaches the programmed value to trigger an interrupt towards the CPU.

On wakeup, due to a sequencer activity, the wake-up block:

- manages the SoC wake-up event generation (to restart the power and clock systems)
- manages the BLE wake-up event (supposed to be done once the power and clocks are ready).

In this case, no IRQ is asserted for signaling the SoC wake-up event.

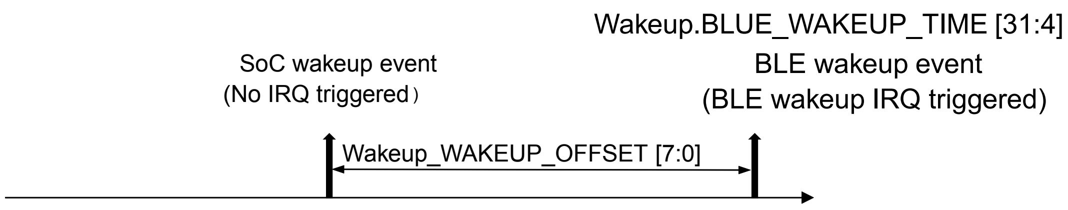
The principle is to wake up the SoC before waking up the BLE to allow time to power and clock to settle. This is possible by programming two pieces of information in the wake-up block:

- BLE wake-up event time:
- Absolute time information to be filled in the Wakeup.BLUE_WAKEUP_TIME[31:0] APB register.
  - This value is in 16 x slow clock periods time units (typically 512 kHz)
- SoC wake-up event time:
  - A wake-up offset information must be filled in the Wakeup.WAKEUP_OFFSET register.
  - This value is in slow clock period time units (typically 32 kHz)

Both SoC and BLE wake-up events generated by the wake-up block are only based on the 32 kHz. Then the wake-up block raises:

- a wake-up request for the power controller when the absolute time reaches the programmed values minus the wake-up offset (anticipated wakeup to settle clock and power before BLE starts). Then, when absolute_time[31:4] is equal to Wakeup.BLUE_WAKEUP_TIME[31:4] minus Wakeup_WAKEUP_OFFSET[7:0]
- a wake-up request for the BLE controller when the absolute_time[31:4] matches the 28 MSB of the programmed value.

Figure 4. **Wake-up event**



A similar feature allows using the existing slow clock timer to generate a CPU wake-up source. This feature, when activated, has no impact on the BLE transfers (no trigger event generated to the BLE sequencer).

In this case, the CPU wake-up process occurs in two steps:

- At ABSOLUTE_TIME[31:4] = CM0_WAKEUP_TIME [31:4] - Wakeup.WAKEUP_OFFSET[7:0]
  The wake-up block raises an SoC wake-up request towards the Power Controller of the SoC to request voltage/clock restoring.
- At ABSOLUTE_TIME[31:4] = CM0_WAKEUP_TIME [31:4]
  If WAKEUP_CM0_IRQ_ENABLE.WAKEUP_IT = 1, the wake-up block sends an interrupt towards the CPU (irq_wakeup_cpu line).

7.4 Wake-up block registers

Table 159. **Wake-up block register list**

Address offset	Name	RW	Reset	Description
0x08	WAKEUP_OFFSET	RW	0x00000000	Wake-up offset register
0x10	ABSOLUTE_TIME	R	0x00000000	Absolute time register
0x14	MINIMUM_PERIOD_LENGTH	R	0x00000000	Minimum period length register
0x18	AVERAGE_PERIOD_LENGTH	R	0x00000000	Average period length register
0x1C	MAXIMUM_PERIOD_LENGTH	R	0x00000000	Maximum period length register
0x20	STATISTICS_RESTART	RW	0x00000000	Statistics restart register
0x24	BLUE_WAKEUP_TIME	RW	0x00000000	BLE wake-up time register
0x28	BLUE_SLEEP_REQUEST_MODE	RW	0x00000007	BLE sleep request mode register
0x2C	CM0_WAKEUP_TIME	RW	0x00000000	CPU wake-up time register
0x30	CM0_SLEEP_REQUEST_MODE	RW	0x80000007	CPU sleep request mode register
0x40	WAKEUP_BLE_IRQ_ENABLE	RW	0x00000000	Wakeup BLE interrupt enable register
0x44	WAKEUP_BLE_IRQ_STATUS	RW	0x00000000	Wakeup BLE interrupt status register
0x48	WAKEUP_CM0_IRQ_ENABLE	RW	0x00000000	Wakeup CPU interrupt enable register
0x4C	WAKEUP_CM0_IRQ_STATUS	RW	0x00000000	Wakeup CPU interrupt status register

Table 160. **WAKEUP_OFFSET register description**

Bit	Field name	Reset	RW	Description
7:0	WAKEUP_OFFSET	0x0	RW	Time to let the power and clock to settle up. This value is in slow clock period time units (typically 32 kHz).
31:8	RESERVED_31_8	0x0	RW	Reserved

Table 161. **ABSOLUTE_TIME register description**

Bit	Field name	Reset	RW	Description
31:0	ABSOLUTE_TIME	0x0	R	Absolute time Unit of this full bit field is (slow_clock *16) frequency period cycle (typically 512 kHz). Note: ABSOLUTE_TIME[31:4] is clocked on the slow clock (typically 32 kHz), ABSOLUTE_TIME[3:0] is the interpolation at slow clock * 16 frequency (typically 512 kHz).

Table 162. **MINIMUM_PERIOD_LENGTH register decription**

Bit	Field name	Reset	RW	Description
3:0	RESERVED3_0	0x0	R	Reserved
13:4	LENGTH	0x0	R	Minimum period length computed by time interpolator
31:14	RESERVED31_14	0x0	R	Reserved

Table 163. **AVERAGE_PERIOD_LENGTH register description**

Bit	Field name	Reset	RW	Description
3:0	LENGTH_FRAC	0x0	R	Additional information/precision on slow clock frequency. Reading AVERAGE_PERIOD_LENGTH[13:0] indicates the number of 16 MHz clock cycles contained in 16 slow clock periods. This bit field is updated every 16 slow clock periods.
13:4	LENGTH_INT	0x0	R	Average period length computed by Time Interpolator. This value indicates the number of 16 MHz clock cycles contained in 1 slow clock period. This bit field is updated every 16 slow clock periods.
23:14	RESERVED23_14	0x0	R	Reserved
31:24	AVERAGE_COUNT	0x0	R	Number of slow clock cycles. This value indicates the number of slow clock periods taken into account to calculate the average. This bit field is updated every slow clock period. This bit field is clamped at 0xFF so reading 0xFF means at least 128 slow clock periods are already being used to calculate the average.

Table 164. **MAXIMUM_PERIOD_LENGTH register description**

Bit	Field name	Reset	RW	Description
3:0	RESERVED3_0	0x0	R	Reserved
13:4	LENGTH	0x0	R	Maximum period length computed by Time Interpolator.
31:14	RESERVED31_14	0x0	R	Reserved

Table 165. STATISTIC_RESTART register description

Bit	Field name	Reset	RW	Description
0	CLR_MIN_MAX	0x0	RW	Write '1' to clear the minimum and maximum registers. Note: This bit is auto cleared by the HW.
1	CLR_AVR	0x0	RW	Write '1' to clear the AVERAGE_PERIOD_LENGTH register value. This action clears both the average length value and the average counter. Note: This bit is auto cleared by the HW.
31:2	RESERVED31_2	0x0	R	Reserved

Table 166. BLE_WAKEUP_TIME register description

Bit	Field name	Reset	RW	Description
31:0	WAKEUP_TIME	0x0	RW	Programmed wake-up time. Unit is in (16 x slow clock) period so typically 512 kHz when slow clock is 32 kHz.

Table 167. BLUE_SLEEP_REQUEST_MODE register description

Bit	Field name	Reset	RW	Description
2:0	RESERVED2_0	0x7	RW	Reserved
28:3	RESERVED28_3	0x0	R	Reserved
29	SLEEP_EN	0x0	RW	- 0: disable BLE IP sleeping mode = no low power mode request when the BLE controller indicates it is no longer busy. - 1: enable BLE IP sleeping mode = low power mode request when the BLE controller indicates it is no longer busy. Note: BLE sequencer is no longer busy if no sequence is on-going and if no Timer1 nor Timer2 counter is enabled (to trig the next sequence).
30	BLE_WAKEUP_EN	0x0	RW	- 0: disable the BLE IP wakeup - 1: enable the BLE IP wake-up request through the embedded wake-up timer. This bit is auto-cleared by hardware when a wake-up event occurs (BLE wake-up time matches with current time).
31	FORCE_SLEEPING	0x0	RW	- 0: the BLE sleeping is managed internally by both BLE IP/wake-up block - 1: the BLE IP is always considered as sleeping by the wake-up block.

Table 168. CM0_WAKEUP_TIME register description

Bit	Field name	Reset	RW	Description
3:0	RESERVED3_0	0x0	R	Always read as zero as no 512 kHz granularity on this time wakeup.
31:4	WAKEUP_TIME	0x0	RW	Programmed wake-up time. Unit is in slow clock period.

Table 169. CM0_SLEEP_REQUEST_MODE register description

Bit	Field name	Reset	RW	Description
2:0	RESERVED2_0	0x7	RW	Reserved
29:3	RESERVED29_3	0x0	R	Reserved
30	CPU_WAKEUP_EN	0x0	RW	- 0: disable/mask the CPU wake-up request.

Bit	Field name	Reset	RW	Description
				- 1: enable the CPU wake-up request. Note: This feature is optional as the CPU is also able to directly manage the sleep / wakeup with the SoC. Main interest is to offer additional features not proposed by the CPU: - an always-on embedded counter able to wake up the CPU at a programmed time.
31	FORCE_SLEEPING	0x1	RW	- 0: the CPU sleeping is managed internally by both the CPU / wake-up block. - 1: the CPU will always be considered as sleeping by the wake-up block.

Table 170. WAKEUP_BLE_IRQ_ENABLE register description

Bit	Field name	Reset	RW	Description
0	WAKEUP_IT	0x0	RW	- 0: disable the BLE wake-up interrupt towards CPU. - 1: enable BLE wake-up interrupt towards the CPU.
31:1	RESERVED31_0	0x0	R	Reserved

Table 171. WAKEUP_BLE_IRQ_STATUS register description

Bit	Field name	Reset	RW	Description
0	WAKEUP_IT	0x0	RW	Write ‘1’ to clear the interrupt. When read, returns the interrupt status.
31:1	RESERVED31_0	0x0	R	Reserved

Table 172. WAKEUP_CM0_IRQ_ENABLE register description

Bit	Field name	Reset	RW	Description
0	WAKEUP_IT	0x0	RW	- 0: disable the CPU wake-up interrupt towards CPU. - 1: enable CPU wake-up interrupt towards the CPU.
31:1	RESERVED31_0	0x0	R	Reserved

Table 173. WAKEUP_CM0_IRQ_STATUS register description

Bit	Field name	Reset	RW	Description
0	WAKEUP_IT	0x0	RW	Write ‘1’ to clear the interrupt. When read, returns the interrupt status.
31:1	RESERVED31_0	0x0	R	Reserved

Revision history

Table 174. **Document revision history**

Date	Revision	Changes
23-Nov-2020	1	Initial release.

Contents

List of tables

List of figures