Digital Baseband-Modem

Version v1.0

Table of Contents

1. Dual-Mode Baseband-Modem . . . 1
   1.1. Architecture Overview . . . 1
   1.2. Baseband-Modem Frontend . . . 1
   1.3. Commands and MMIO Registers . . . 2
   1.4. List of Commands . . . 4
   1.5. Configuring Modem LUTs . . . 6
   1.6. Configuring the FIR filter coefficients . . . 7
   1.7. Interrupts . . . 8
   1.8. IEEE 802.15.4 LR-WPAN Baseband . . . 8
   1.9. Bluetooth Low Energy (BLE) Baseband . . . 10
      1.9.1. Whitening . . . 11
   1.10. Unified FSK Modem . . . 11
      1.10.1. TX Chain . . . 11
      1.10.2. RX Chain . . . 12
2. Example Radio Front-End (RFE) . . . 15
   2.1. Specification . . . 15
   2.2. Example On-Chip RFE Architecture . . . 15

1. Dual-Mode Baseband-Modem

The Dual-Mode Baseband-Modem (BM) is responsible for handling the transmission and reception of both Bluetooth LE and IEEE 802.15.4 packets. For Bluetooth Low Energy, the BM handles Bluetooth Low Energy 1M Uncoded Link Layer Packets. For IEEE 802.15.4, the BM handles packets transmitted according to the PHY standard for the O-QPSK modulation in the 2.4 GHz band. Additionally the BM exposes an interface for the CPU to read/write various information (e.g. tuning bits) to the analog RF.

There are two primary sub-components in the Baseband-Modem, the Baseband, and the Modem. The baseband is responsible for the bit-stream processing of incoming and outgoing packets. The modem is responsible for modulation on the TX side, and digital image rejection and demodulation on the RX side.

1.1. Architecture Overview

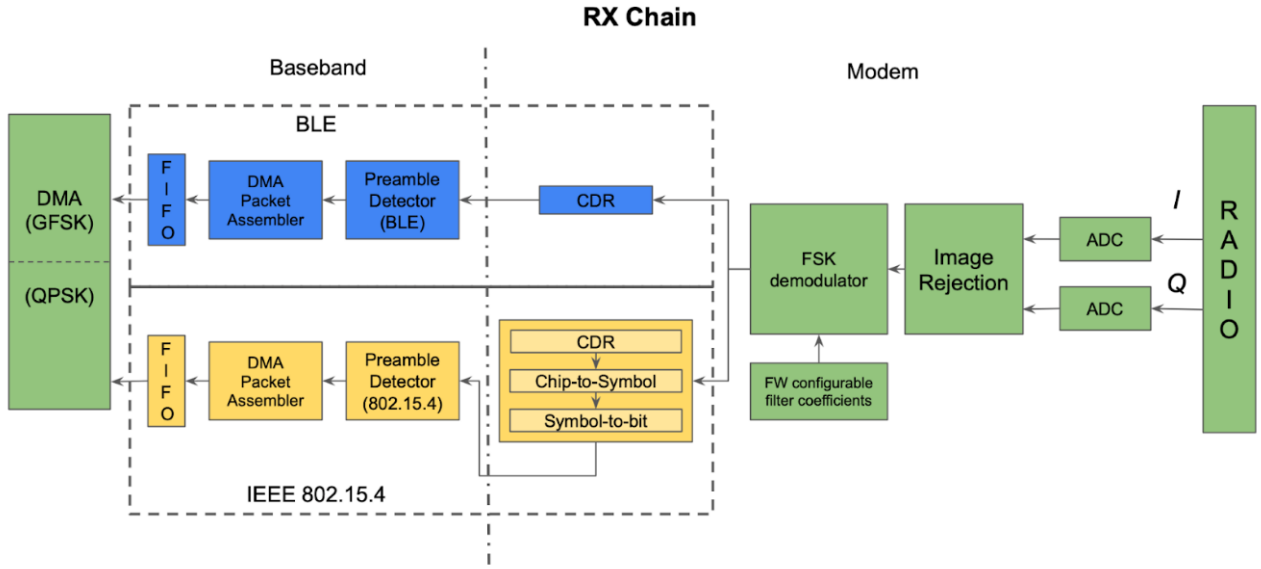

Figure 1. Receive chain system block diagram.

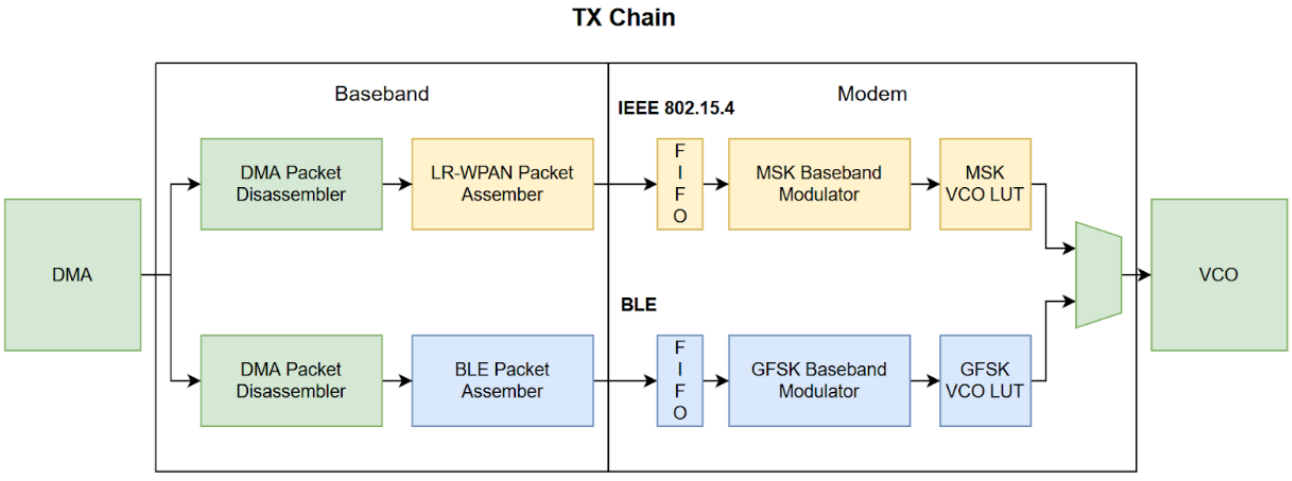

Figure 2. Transmit chain system block diagram.

1.2. Baseband-Modem Frontend

1.3. Commands and MMIO Registers

The baseband-modem (BM) block contains a set of memory mapped registers used for passing commands to the BM, tuning analog RF components, and making BM status visible to the CPU. The address map is shown below. Note that the addresses are relative to the BM's base attachment address. For example, if the BM is attached at 0x8000, then the address of 0x04 corresponds to 0x8004.

ADDR	DATA	Size (bits)	Description
0x00	Instruction	32	Instruction received from processor. Contains 4 bits of primary instruction, 4 bits of secondary instruction, then 24 bits of data.
0x04	Additional Data	32	Additional data to write. Set data before writing instructions when applicable.
0x08	Status 0	32	[2:0] Assembler State [5:3] Disassembler State [7:6] TX State [10:8] RX Controller State [12:11] TX Controller State [15:13] Controller State [23:16] ADC I data [31:24] ADC Q data
0x0C	Status 1	32	[5:0] Modulation LUT index [10:6] I AGC LUT index [15:11] I DCOC LUT index [20:16] Q AGC LUT index [25:21] Q DCOC LUT index
0x10	Status 2	32	[31:0] BLE CDR bit count
0x14	Status 3	32	[31:0] LRWPAN CDR bit count
0x18	Status 4	32	[31:0] LO/32 counter
0x1C	General trim 0	8	General trim bits (N/C)
0x1D	General trim 1	8	[0] LO/32 input (external, not implemented)
0x1E	General trim 2	8	General trim bits (N/C)
0x1F	General trim 3	8	General trim bits (N/C)
0x20	General trim 4	8	General trim bits (N/C)
0x21	General trim 5	8	General trim bits (N/C)
0x22	General trim 6	8	General trim bits (N/C)
0x23	General trim 7	8	Debug Enable (0 = debug enable)
0x24	I VGA gain control	10	Manual VGA value if not using I AGC
0x26	I VGA attenuation reset	1	reset the I AGC
0x27	I VGA attenuation useAGC	1	use I AGC (manual VGA value if not)
0x28	I VGA attenuation sample window	8	I AGC sample window
0x29	I VGA attenuation ideal peak to peak	8	I AGC ideal peak to peak
0x2A	I VGA tolerance peak to peak	8	I AGC peak to peak tolerance
0x2B	I BPF CHP 0 & 1	8	I bandpass filter tuning
0x2C	I BPF CHP 2 & 3	8	I bandpass filter tuning
0x2D	I BPF CHP 4 & 5	8	I bandpass filter tuning

ADDR	DATA	Size (bits)	Description
0x2E	I BPF CLP 0 & 1	8	I bandpass filter tuning
0x2F	I BPF CLP 2	4	I bandpass filter tuning
0x30	Q VGA gain control	10	Manual VGA value if not using Q AGC
0x32	Q VGA attenuation reset	1	reset the Q AGC
0x33	Q VGA attenuation useAGC	1	use Q AGC (manual VGA value if not)
0x34	Q VGA attenuation sample window	8	Q AGC sample window
0x35	Q VGA attenuation ideal peak to peak	8	Q AGC ideal peak to peak
0x36	Q VGA tolerance peak to peak	8	Q AGC peak to peak tolerance
0x37	Q BPF CHP 0 & 1	8	Q bandpass filter tuning
0x38	Q BPF CHP 2 & 3	8	Q bandpass filter tuning
0x39	Q BPF CHP 4 & 5	8	Q bandpass filter tuning
0x3A	Q BPF CLP 0 & 1	8	Q bandpass filter tuning
0x3B	Q BPF CLP 2	4	Q bandpass filter tuning
0x3C	I DCO use	1	toggle using I DCO
0x3D	I DCO reset	1	reset the I DCO
0x3E	I DCO gain	8	set gain for I DCO (unsigned FixedPoint w/ 2 bit binary point)
0x3F	Q DCO use	1	toggle using Q DCO
0x40	Q DCO reset	1	reset the Q DCO
0x41	Q DCO gain	8	set gain for Q DCO (unsigned FixedPoint w/ 2 bit binary point)
0x42	DCOC tuning 1	6	Manual I current DAC for stage 2 VGA value if not using I DCO
0x43	DCOC tuning 2	6	Manual Q current DAC for stage 2 VGA value if not using Q DCO
0x46	MUX debug in	10	Debug configuration, input
0x48	MUX debug out	10	Debug configuration, output
0x4A	Enable RX I	5	Manual enable RX I values {3'b0, mix, buf, vga_s1, vga_s2, bpf}
0x4B	Enable RX Q	5	Manual enable RX Q values {3'b0, mix, buf, vga_s1, vga_s2, bpf}
0x4C	Enable VCO LO	2	Manual enable VCO LO {6'b0, vco_lo, lna}
0x50	LUT command	32	LUT set instruction [3:0] LUT ID [9:4] address (index) [31:10] value
0x54	RX error message	32	Interrupt message, RX error message
0x58	RX finish message	32	Interrupt message, RX finish message
0x5C	TX error message	32	Interrupt message, TX error message
0x60	FIR command	32	FIR filter reprogramming instruction, [3:0] FIR ID [9:4] coefficient (index) [31:10] value
0x64	I VGA attenuation gain increase	8	I AGC gain increase step size (by LUT index)
0x65	I VGA attenuation gain decrease	8	I AGC gain decrease step size (by LUT index)
0x66	Q VGA attenuation gain increase	8	Q AGC gain increase step size (by LUT index)

ADDR	DATA	Size (bits)	Description
0x67	Q VGA attenuation gain decrease	8	Q AGC gain decrease step size (by LUT index)

In order to pass commands to the BM, two 32-bit registers at addresses 0x00 and 0x04 are utilized. The register at 0x04 contains the additional data field for a given command while the register at 0x00 contains the instruction. Writing to the register at 0x00 is the trigger for the BM to execute a given command. Accordingly, **the processor should always write to the additional data register prior to writing to the instruction register for a given command.** The format for a BM instruction and a list of available instructions are detailed below.

Bits	31-8	7-4	3-0
Field	Data	Instruction 2	Instruction 1

1.4. List of Commands

Configure Command

Configure baseband constants. The constant is selected using the instruction 2 field and set to the value specified in the additional data field. In the case that the secondary instruction is set to CONFIG_LUT, reference the LUT addresses provided in the explanations below/

Field	Data	Instruction 2	Instruction 1
Value	X unless CONFIG_LUT, then LUT address	See table	0

Field	Additional Data
Value	Value for the constant to be set to

Instruction 2	Name
0x0	CONFIG_RADIO_MODE
0x1	CONFIG_CRC_SEED
0x2	CONFIG_ACCESS_ADDRESS
0x3	CONFIG_SHR
0x4	CONFIG_BLE_CHANNEL_INDEX
0x5	CONFIG_LRWPAN_CHANNEL_INDEX

Configuration instruction descriptions

(0x0) CONFIG_RADIO_MODE

Specify the following in the Additional Data register (0x04) prior to issuing the instruction 0: Set the radio transceiver mode to Bluetooth Low Energy 1: Set the radio transceiver mode to IEEE 802.15.4 LR-WPAN Valid in any radio mode.

(0x1) CONFIG_CRC_SEED

Set the CRC (cyclic redundancy check) seed for the BLE and LR-WPAN CRC circuits to the value in the Additional Data register (0x04). This value changes for BLE and should be 0 for LR-WPAN

(0x2) CONFIG_ACCESS_ADDRESS

Set the BLE Access Address for the BLE uncoded packet. This value must be provided by the CPU. More information may be found in the BLE Baseband section. NOTE: An access address of 0xFFFFFFF6 disables whitening of transmitted BLE packets

(0x3) CONFIG_SHR

Set the LRWPAN SHR to match with for receiving a LRWPAN packet. This value must be provided by the CPU. According to 802.15.4 spec, it should be a fixed value of 0xA700 (only 2 bytes are matched due to lower hardware cost), but is programmable for flexibility.

(0x4) CONFIG_BLE_CHANNEL_INDEX

Values for the channel index can range from 0 to 39, corresponding to BLE channel frequencies beginning at 2402 MHz until 2480 MHz. It is critical to note that the channel index from 0 to 39 is a direct mapping to channel frequencies - NOT the BLE channel numbering scheme that considers advertising channels separately.. For example, setting CONFIG_BLE_CHANNEL_INDEX to 0 will result in a transmission with center frequency at 2402 MHz. This corresponds to BLE channel 37. More information: https://www.rfwireless-world.com/Terminology/BLE-Advertising-channels-and-Data-channels-list.html

(0x5) CONFIG_LRWPAN_CHANNEL_INDEX

Values for the channel index can range from 0 to 15, corresponding to LR-WPAN channel frequencies beginning at 2405 MHz until 2480 MHz.

Send Command

Transmit a specified number of PDU header and data bytes. Bytes are retrieved by the BM by loading them from the specified address.

Field	Data	Instruction 2	Instruction 1
Value	Number of bytes	X	1

Field	Additional Data
Value	Load address

Receive Enter Command

Place the device into receive mode. If a message is picked up, it will be stored starting at the specified address address.

Field	Data	Instruction 2	Instruction 1
Value	X	X	2

Field	Additional Data
Value	Storage address

Receive Exit Command

Exit the device from receive mode. This command will succeed as long as the device has not yet matched an instruction preamble.

Field	Data	Instruction 2	Instruction 1
Value	X	X	3

Field	Additional Data
Value	X

Debug Command

Turns on both the TX and RX paths according to the specified loopback mask before passing the specified

number of PDU bytes in a loop. For simplicity the return data is stored at <load address + total bytes> rounded to the next byte aligned address. The loopback mask is used to determine at which point the data should be revered and sent back towards the CC.

Field	Data	Instruction 2	Instruction 1
Value	Total bytes to send	Loopback mask	15

Field	Additional Data
Value	Load address and base for store address calculation

Loopback Mask	Loopback Point
0b0001 (0x1)	Empty
0b0010 (0x2)	In the modem, loop the FSKTX output back in to the CDR in FSKRX
0b0100 (0x4)	Empty
0b1000 (0x8)	Empty

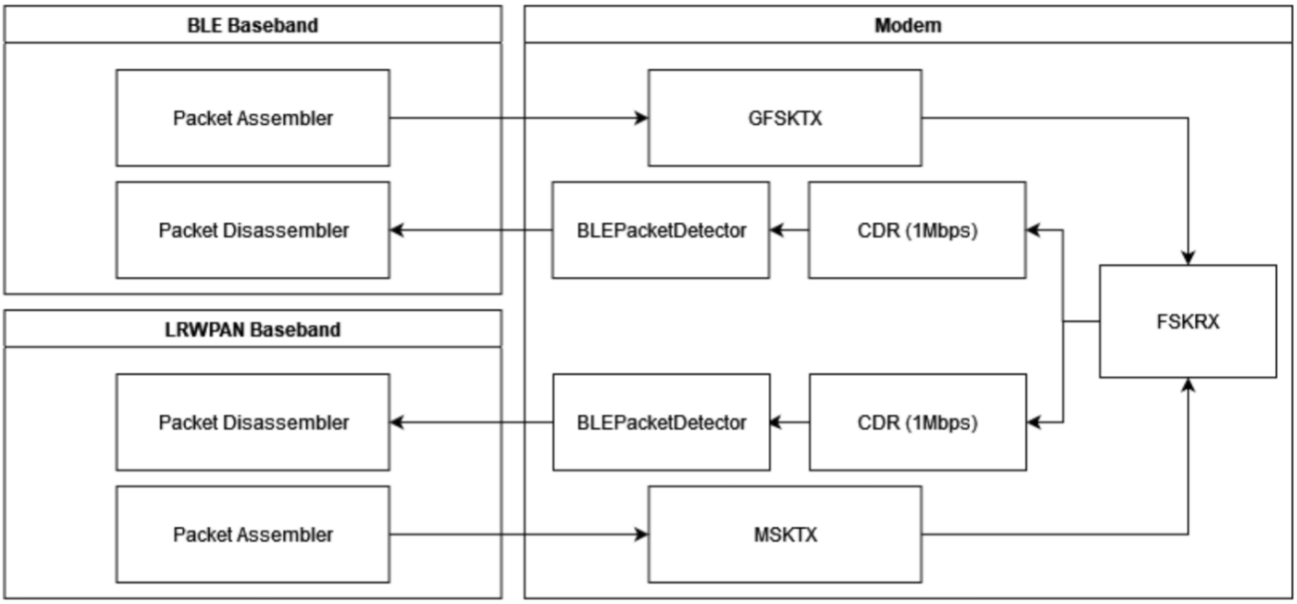

Figure 3. Loopback diagram, note that the FSKRX module handles loopback logic and directs the output of MSKTX & GFSK TX modules to respective CDR blocks

1.5. Configuring Modem LUTs

The modem LUTs can be configured using the LUT command register in the MMIO map enumerated above (register offset 0x50).

LUT IDs are defined as follows:

0x0 - VCO_MOD

64-entry, 8-bit valued LUT that is sampled to produce frequency deviations from the center frequency. This LUT must be reloaded on each channel and mode change. 0 indicates the largest negative frequency deviation, 63 indicates the largest positive deviation, and 31 indicates no frequency deviation. Note that 0/63 should be +/-250kHz for BLE and +/-500kHz for LR-WPAN.

0x1 - VCO_CT_BLE

40-entry LUT sampled with BLE channel index to produce coarse and medium tuning bits to drive the VCO to the center frequency of the BLE channel index selected. Does not apply when in LR-WPAN radio mode.

0x2 - VCO_CT_LRWPAN

16-entry LUT that is sampled with the LR-WPAN channel index to produce the coarse and medium tuning bits to drive the VCO to the center frequency of the LR-WPAN channel index selected. Does not apply when in BLE radio mode.

0x3 - AGC_I

64-entry LUT that is sampled to drive the VGA attenuation gain for the I channel VGA. Only the lower 32 entries are used.

0x4 - AGC_Q

64-entry LUT that is sampled to drive the VGA attenuation gain for the Q channel VGA. Only the lower 32 entries are used.

0x5 - DCO_I

64-entry LUT that is sampled to drive the current DAC for the I channel stage 2 VGA. Only the lower 32 entries are used.

0x6 - DCO_Q

64-entry LUT that is sampled to drive the current DAC for the Q channel stage 2 VGA. Only the lower 32 entries are used.

1.6. Configuring the FIR filter coefficients

The FIR filter coefficients can be configured using the FIR command register in the MMIO map enumerated above (register offset 0x60).

FIR IDs are defined as follows:

0x0 - NONE

Reserved

0x1 - TX_GAUSSIAN

24-entry, 8-bit FixedPoint (2-bit fractional component) filter for BLE GFSK. Only 16 coefficients are used. Note the filter was designed to run at 8MHz (length = 2*cycles/bit) but there's a bug in the implementation. The short-term fix to get it working was to double the coefficients.

0x2 - RX_HILBERT_I

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for image rejection.

0x3 - RX_HILBERT_Q

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for image rejection. The output of the I and Q filters are summed to form the image rejected signal suitable for FSK demodulation.

0x4 - RX_MATCH_SIN_F0

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for matched filter for binary 0. Two templates (filters) are used to account for phase shift. Note BLE coefficients are loaded by default. LR-WPAN ones must be loaded separately.

0x5 - RX_MATCH_COS_F0

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for matched filter for binary 0. Two templates (filters) are used to account for phase shift. Note BLE coefficients are loaded by default. LR-WPAN ones must be loaded separately.

0x6 - RX_MATCH_SIN_F1

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for matched filter for binary 1. Two templates

(filters) are used to account for phase shift. Note BLE coefficients are loaded by default. LR-WPAN ones must be loaded separately.

0x7 - RX_MATCH_COS_F1

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for matched filter for binary 1. Two templates (filters) are used to account for phase shift. Note BLE coefficients are loaded by default. LR-WPAN ones must be loaded separately.

0x8 - RX_LPF

32-entry, 16-bit FixedPoint (12-bit fractional component) filter for bitrate filter of the matched filter output. Two filters with the same coefficients are used, one for 0s and another for 1s. Note BLE coefficients are loaded by default. LR-WPAN ones must be loaded separately.

1.7. Interrupts

The Baseband-Modem provides five interrupts as outputs from the Baseband Frontend. These interrupts provide the CPU with triggers for events from the transceiver module. The interrupts are described in the table below:

Table 1. Baseband-Modem interrupt table

Interrupt Index	Name	Description
0	RX_ERROR	Triggers on recognition of an error while the BM is in RX state. The MMIO register 0x54 will be populated with an RX_ERROR_MESSAGE. Presently not implemented.
1	RX_START	Triggers when the baseband disassembler/correlator enters the busy state. Denotes that a packet has been received and demodulated, but has not yet been disassembled.
2	RX_FINISH	Triggers when the disassembler has completed processing an incoming message. Populates the MMIO register RX_FINISH_MESSAGE at 0x58 with the length of the message.
3	TX_ERROR	Triggers on recognition of an error while the BM is in TX state. The MMIO register 0x5C will be populated with an TX_ERROR_MESSAGE. Presently not implemented.
4	TX_FINISH	Triggers when the modem has completed transmission of all bytes in the provided message. The DMA read response bytes must match the number of bytes specified in the Send Command for this interrupt to fire.

1.8. IEEE 802.15.4 LR-WPAN Baseband

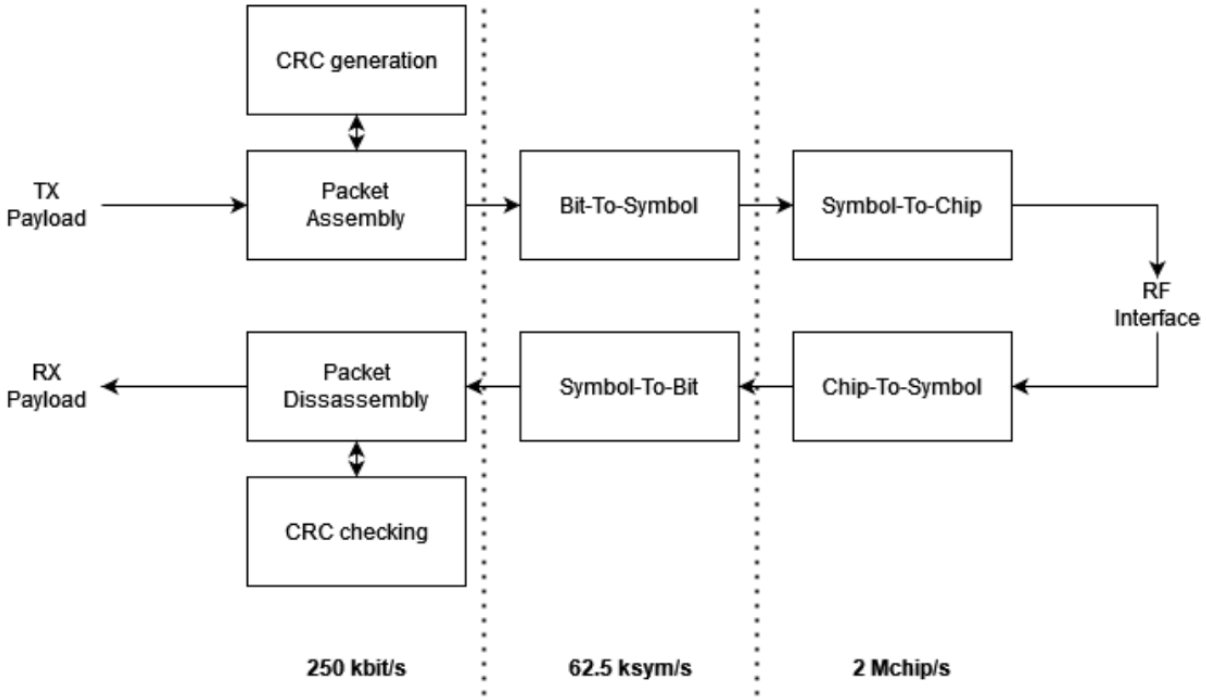

Figure 4. LR-WPAN baseband and modulation/demodulation diagram

The LR-WPAN baseband transmission chain involves DMA packet disassembly, LR-WPAN packet assembly, CRC generation, bit-to-symbol translation, and symbol-to-chip translation before the final chip stream is provided to the modem for modulation. The packet is assembled according to the IEEE 802.15.4 (LR-WPAN) PPDU schematic (shown below).

It is important to note the rates at each stage of the baseband TX chain. The standard calls for a 250 kb/s bitrate corresponding to packet assembly/packet disassembly. Following bit-to-symbol translation, the symbol rate becomes 62.5 ksym/s (where 4 bits specifies 1 symbol). Following symbol-to-chip translation, the chip rate now matches the LR-WPAN nominal baseband frequency of 2 MHz frequency or 2 Mchip/s (where 1 symbol specifies 32 chips). The inverse applies for the RX chain.

During packet assembly, a frame check sequence is calculated as a 16-bit ITU-T CRC.

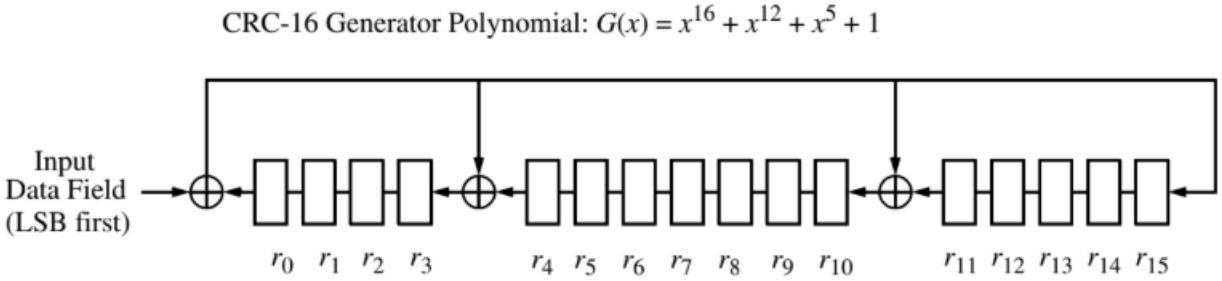

Figure 5. CRC calculation diagram

The algorithm for the CRC begins by setting all remainder registers, r0…r15, to zero. Next, we start from the LSB and shift the MHR and payload into the divider. The FCS is then the remainder register after the last bit of the data is shifted in. Lastly, the FCS is appended to the data field so that register r0 is transmitted first.

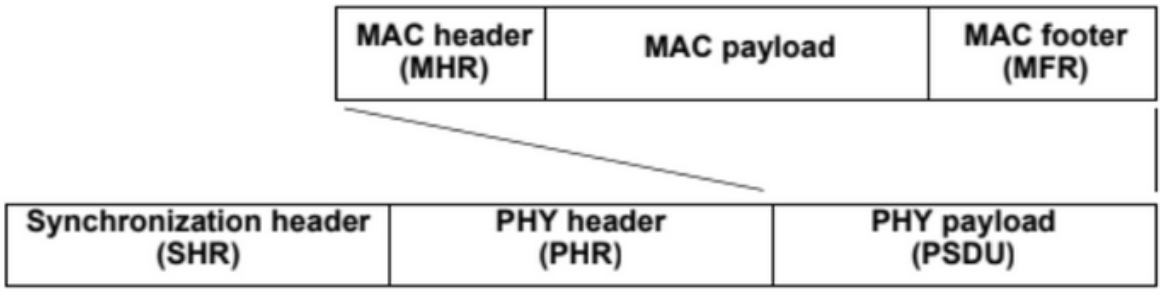

Figure 6. IEEE 802.15.4 PPDU Schematic

The frame check sequence (FCS) is appended to the end of the PPDU as the MAC footer (MFR) in the diagram above. If the result of CRC checking is false, the [flag] is set to high, before triggering a RX_ERROR message.

PSDU Schematic

The preamble for 15.4 is a 32-bit sequence of all zeros. The SFD is loosely analogous to the Access-Address (AA) of the BLE Baseband in that they both uniquely identify the start of the packet for the protocol used.

1.9. Bluetooth Low Energy (BLE) Baseband

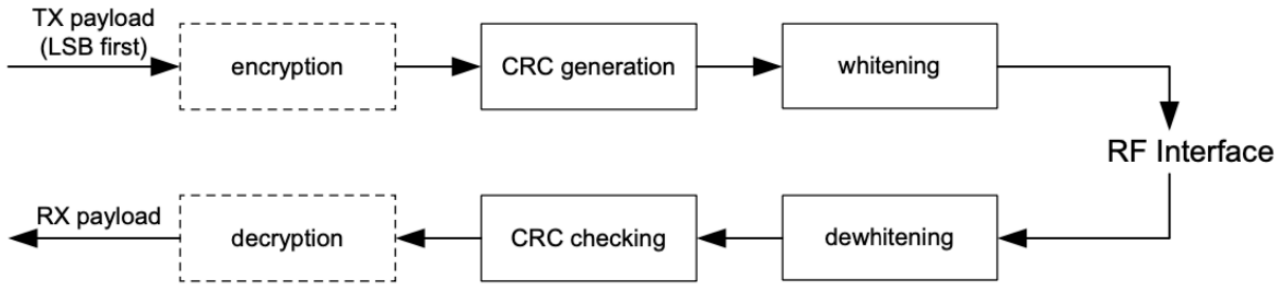

Figure 7. BLE baseband and modulation/demodulation diagram

Below is the BLE Uncoded Packet format. All elements of the packet either are or depend on data that the CPU must set (which is taken care of by the SW implementation of the Link Layer). These values are communicated to the packet assembler and disassembler through the controller via the command system over MMIO.

Figure 8. BLE PPDU Schematic

The Preamble is an alternating series of 1s and 0s that depends on the LSB of the Access-Address (AA). The CPU need not provide any additional control signals for this part of the packet.

The Access-Address is a special value set by the SW Link Layer that determines whether a packet is valid or not. This value must be provided by the CPU and depends on past communications or may be set to some default value.

The PDU is the already encrypted content of the packet, the CPU must provide this in its entirety. Through the DMA controller, the packet assembler will retrieve this from main memory. The packet disassembler will deliver a PDU to the main memory once one is received.

The CRC (cyclic redundancy check) is derived from the Link Layer state (which must be provided by the CPU via MMIO register). In the receive chain, if the calculated CRC does not match the CRC found in the PDU, the [flag]

is set to high.

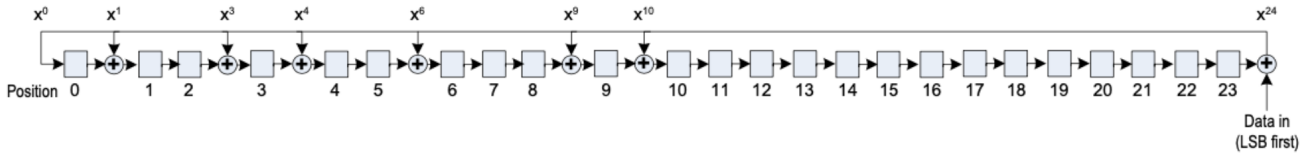

Figure 9. BLE CRC functional schematic

1.9.1. Whitening

In order to avoid long continuous sequences of 1s and 0s in the transmitted data, the entire packet goes through a process known as whitening.

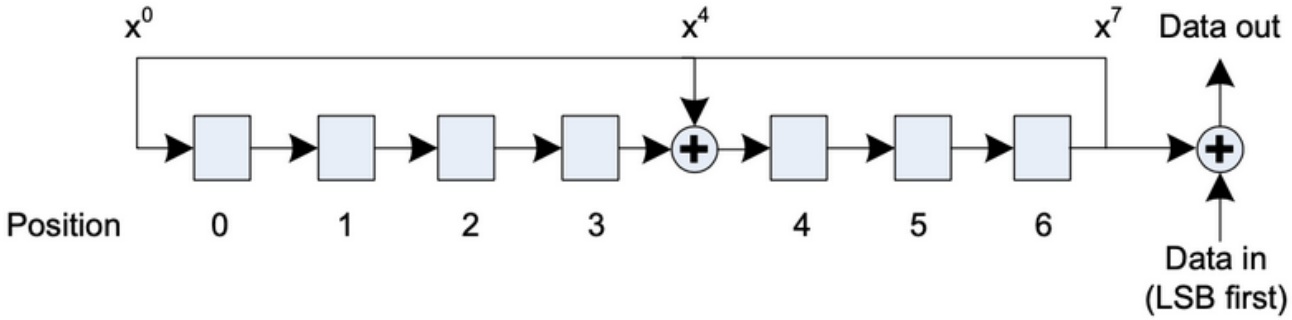

Figure 10. Whitening diagram

Outgoing data on the TX chain is whitened and incoming data is also “de-whitened”

1.10. Unified FSK Modem

The modem sits between the digital baseband and the ADCs / DAC to analog RF. It is responsible for performing demodulation and modulation and clock and data recovery at a rate of 1Mbps for BLE or 2Mchip/s for LR-WPAN. Each cycle, the digital baseband either receives a new bit from the demodulator, or provides a new bit for transmission to the modulator.

1.10.1. TX Chain

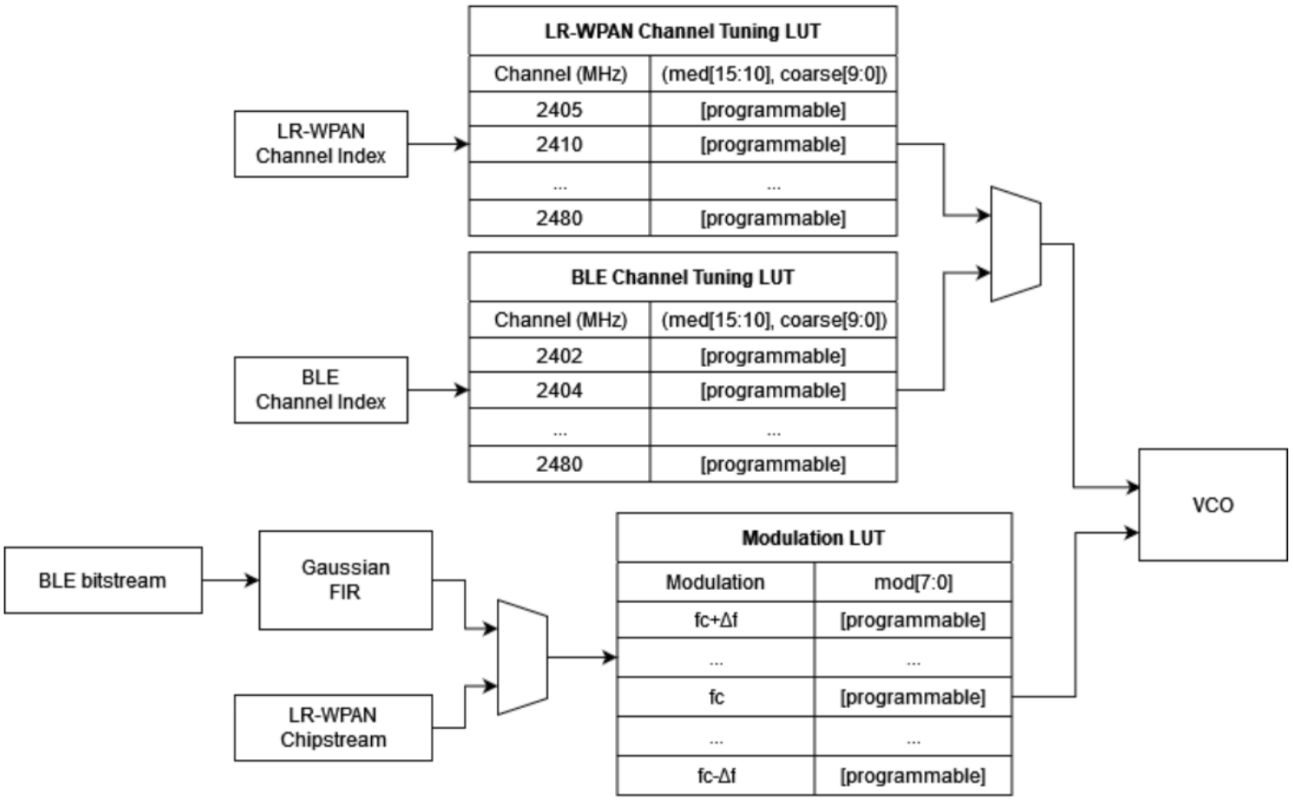

Figure 11. Modem TX chain schematic

The digital baseband bit-stream process outputs individual bits into a FIFO that will be consumed by the modem for use in modulation. For BLE transmissions, the bitstream is modulated using Gaussian frequency shift keying (GFSK). The modulator modulates the outgoing bits using at a rate of 1Msym/s, with a modulation index of 0.5. For LR-WPAN transmissions, the chipstream directly drives the modulation LUT. The output of the modulation LUT is an 8-bit unsigned integer value that indicates the FSK modulation away from the maximum frequency at that channel tuning.

The 6-bit FSK modulation output, in conjunction with a 6 bit channel select value, are used to address into the Modulation LUT to retrieve input values for the VCO LO based on the channel index and modulation frequency offset. These decoding LUTs provide 8-bit values for LO coarse tuning, LO FSK, and the PLL division.

1.10.2. RX Chain

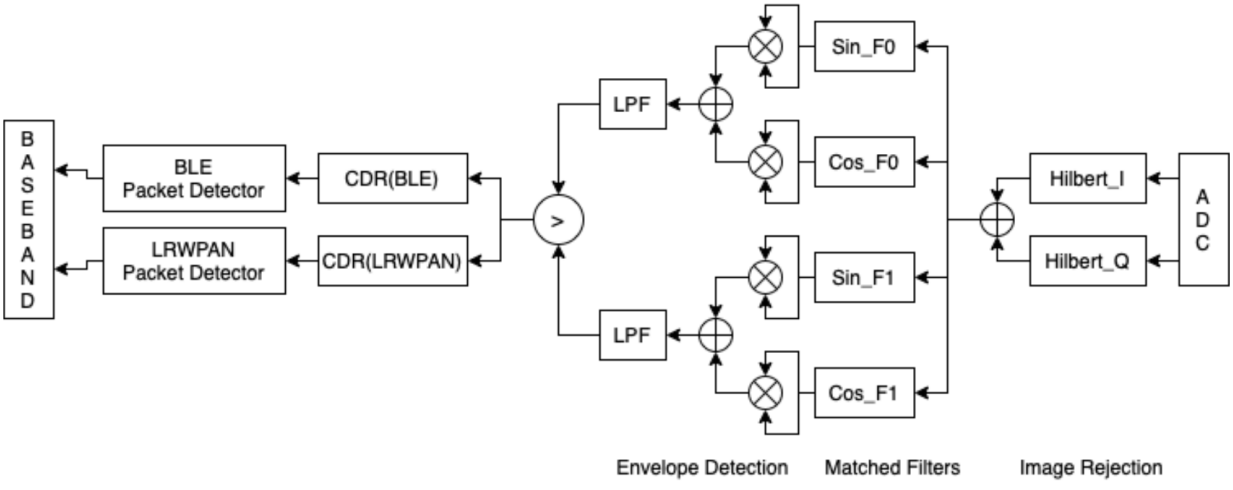

Figure 12. Modem RX chain schematic

After mixing, the analog I and Q signals go through ADCs to produce 8 bit output values. The ADCs are clocked on the negative edge of the digital system clock. On the falling system clock edge, the ADCs sample and latch the signals coming from analog RF. These sampled 8-bit values are then captured by rising-edge triggered flip-flops at the analog-digital boundary of the RX chain.

Image Rejection

After mixing with the LO and being sampled by the I/Q ADCs, the Q signal for the signal (F_LO+F_IF) ends up +90° shifted from I while the image (F_LO-F_IF) ends up -90° shifted from I. A Hilbert filter shifts positive frequencies by +90° and negative frequencies by -90°. By applying it to the I signal and adding to the Q signal, this causes constructive interference for the desired signal and destructive interference for the image, heavily attenuating it. Note these filters are fully configurable (up to their maximum length) if needed.

FSK Demodulation

After image rejection, the matched filters (the optimal non-coherent demodulator!) and envelope detection serve to discriminate between positive and negative frequency deviations. A matched filter performs a convolution of the input signal with a “template” which is simply samples of the desired frequency up to the length of a bit/chip period. Due to phase shift, we must have sine and cosine templates and sum the two. Since the filters generate high frequency noise, we perform “envelope detection” after the matched filters to remove it and get a clean signal giving the magnitude of the match. Comparing the magnitude of each gives us the decision for the deviation.

For an ADC clock of 32MHz, we need a 32-length filter for BLE and 16-length filter for LR-WPAN. For an IF of 2MHz, we need templates of 1.75MHz and 2.25MHz for BLE and 1.5MHz and 2.5MHz for LR-WPAN.

Clock and Data Recovery (CDR)

After FSK demodulation, we have an oversampled bitstream which must downsample to the appropriate bitrate. The goal of the CDR is to align the sampling window with the phase of the incoming bitstream.

Our design has 3 integrators - early, present, late - which each look at a window (of length equal to the bit period) of the incoming bitstream but are offset from each other by 1 cycle. We then have a counter that decrements every cycle. When the counter hits 0, we return a sampled bit that is the value of the strongest integrator (determined by majority and number of 1s and 0s). We then reset the counter to a value that will cause the winning integrator in the next bit period to be closer to the present integrator (either by extending or shortening the amount of time to the next sample time).

For LR-WPAN, the long length of the preamble and chip spreading make slow phase aligning CDRs a non-issue. For BLE, we only have the preamble of 8 symbols to align. For a 32MHz clock and BLE’s 1Mbps, we would be at most 16 cycles off in phase alignment. However, since our CDR always returns the integrator of greatest absolute value rather than the present one each time, it should confidently return the correct symbol despite a large phase shift and it still aligning to an incoming signal.

This idea can easily be extended to more integrators to align to phase shifts more quickly or control the granularity of phase shifts, but we found that 3 integrators and a 1 cycle shift between them worked well in simulation. If our clock is particularly off, the CDR will fail to account for both the phase shift and clock inaccuracy which would necessitate the extension.

Packet Detection

For BLE, the preamble is intended to lock in the AGC and CDR while the following access address is what determines the start of the packet and the byte alignment. The BLE packet detector searches for the access address after which it can confidently send (whitened) bytes over to the baseband. The baseband asserts an end-of-packet signal when it’s done consuming the bytes of the packet.

For LR-WPAN, the preamble again locks in the AGC and CDR while the SFD (0xA7) denotes the start of the packet. Since each 4-bit symbol (within each 8-bit byte) is sent out as a 32-chip sequence, we perform the chip

sequence to symbol mapping based on Hamming distance before looking for the SFD. To minimize false matches while reducing hardware cost, we also match one byte of the preamble (searching for 0xA700) to determine the start of the packet. After that, the packet detector can again confidently send bytes to the baseband.

Automatic Gain Control (AGC)

To support high dynamic range and also keep our ADCs near saturation to maximize demodulation accuracy and SNR, we have an AGC for I and Q. It is a simple design, keeping track of the difference between the minimum and maximum values of the ADC for a given window (which should be long enough to see a sine wave minimum and maximum). If the value exceeds the desired value plus some tolerance, we increase the gain. If it's below, we decrease the gain.

For stable output, keep the tolerance larger than the gain resolution. For fast locking to incoming signals (where the AGC will be at its maximum due to just seeing noise), have the decrement amount be larger than the increment amount.

DC Offset Compensation (DCO)

To remove the DC offset in the ADC output for I and Q, we have a DCO. It simply integrates the incoming signal (interpreted as a FixedPoint between -1 and ~1), applies a gain, and feeds that back into a LUT that will calibrate the VGA to remove the offset.

2. Example Radio Front-End (RFE)

2.1. Specification

The radio must be able to operate in the frequency range of 2.4G to 2.4835 GHz covering 70 1MHz RF channels. In this range, the receive chain should function over an input power range from -70dBm to -10dBm and achieve a BER of 0.1% (translates to an SNR of 12.5dB) in following in band test conditions when the input is at -67dBm, shown below.

Table 2. In-Band Interference, Nominal Temperature

Interference Channel	Ratio (dB)	Input Signal (dBm)	Interferer (dBm)
Co-channel	11	-60 dBm	-71 dBm
Adj. 1 MHz	0	-60 dBm	-60 dBm
Adj. 2 MHz	-30	-60 dBm	-30 dBm
Adj. >3 MHz	-40	-67 dBm	-27 dBm
Image	-9	-67 dBm	-58 dBm
Image +/- 1 MHz	-20	-67 dBm	-47 dBm

Table 3. Out-of-Band Interference

Interference Channel	Interferer (dBm)
30 MHz - 2000 MHz	-10 dBm
2000 MHz - 2399 MHz	-27 dBm
2484 MHz - 3000 MHz	-27 dBm
3000 MHz - 12.75 GHz	-10 dBm

2.2. Example On-Chip RFE Architecture

The example radio front-end architecture below uses a low-IF receiver architecture and a direct modulation transmitter architecture. This architecture is the on-chip RFE that was used during the development of this baseband-modem processor, so most of the analog tuning and configuration registers are specific to this architecture.

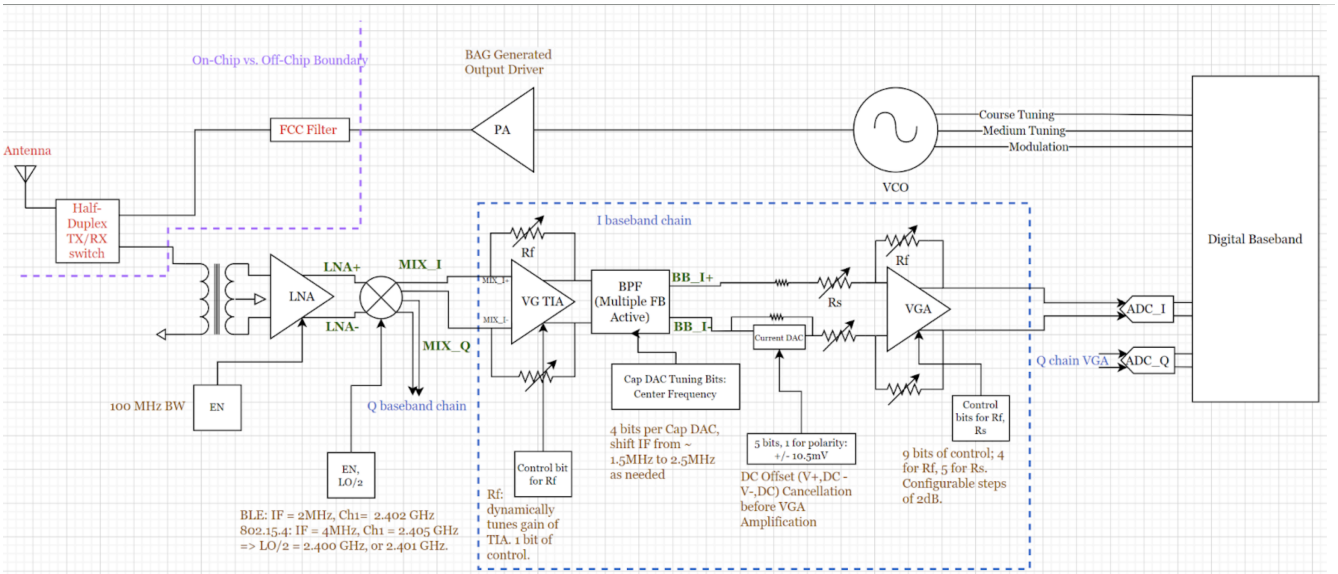

Figure 13. Example on-chip RFE architecture

The baseband-modem's receive chain supports automatic gain control based on tuning the the VGA and VG-TIA gain stages. And its DC offset correction (DCOC) is based on tuning the current DAC at the input of the final VGA stage.

The baseband-modem's transmit modulation implementation is specific to this direct modulation architecture, where the coarse/med/mod bits drive the capacitively tuned VCO.