

SCuM-V23 Specification

Version v1.0

Table of Contents

1. Resource Index	1
2. Overview	2
2.1. Block Level Diagram + Chip Features	2
2.2. 32-bit RISC-V Core	2
2.3. On-Chip Memory System	3
2.4. Busses	3
2.5. Interrupts	3
2.6. Dual-Mode BLE/IEEE 802.15.4 Transceiver	3
3. Power System	4
3.1. Switched Capacitor Converter	4
3.2. Low Dropout (LDO) Voltage Regulators	6
3.3. Current Reference	6
3.4. Bandgap Reference	6
3.4.1. Core BGR Circuit	7
3.4.2. 2-Stage Op-Amp	7
3.5. Utility Current DAC	7
4. Oscillators	8
5. Digital Core	9
5.1. Key Features	9
5.2. Power Consumption	9
5.3. Boot Process	10
6. Digital Baseband-Modem	11
6.1. Dual-Mode Baseband-Modem	11
6.1.1. Architecture Overview	11
6.1.2. Baseband-Modem Frontend	11
6.1.3. Interrupts	18
6.2. IEEE 802.15.4 LR-WPAN Baseband	18
6.3. Bluetooth Low Energy (BLE) Baseband	19
7. Encryption Modules	20
7.1. ECC	20
7.2. AES	20
7.3. SHA-256	21
8. 2.4 GHz Tranceiver	22
8.1. Radio Architecture	22
8.2. Radio Final Block Diagram	22
8.3. All Interface Pins	22
8.4. Transmitter	23
8.4.1. DCO Performance	23
8.4.2. PA Performance	23
8.5. Design Flow, Tradeoffs, and Decisions	23
8.5.1. RF Frontend	24
8.5.2. Baseband	24
8.6. Simulation	25
9. 94 GHz FMCW Radar Transmitter	26
9.1. System Overview	26
9.2. Ramp Generator	26

9.3. R-2R DAC	27
9.4. VCO	27
9.5. Frequency Doubler	28
9.6. PA	28
9.7. Register Map	28
9.8. Layout	29
10. uV-Precision Delta-Sigma ADC	31
10.1. System-level Diagram and Description	31
10.2. Analog Front-End	31
10.3. Digital Front-End	31
10.4. Register Map	32
11. Electrical Characteristics	33
12. Usage Scenarios	34
13. Revision History	35
14. Appendix	36

1. Resource Index

Resource	Description	Link
SCuM-V23 End of Semester Review	Final design review from Sp23	Google Slides
SCuM-V23 Mid-Semester Review	Mid-semester design review from Sp23	Google Slides
Analog RF Simulations	Locations and information on running RF analog simulations	Google Doc
SCuM-V22 Bringup Status	Archive of SCuM-V22 bringup measurements	Google Slides

2. Overview

This document describes the Single-Chip Micro Mote V (SCuM-V), a 16nm system-on-chip (SoC) for internet-of-things (IoT) applications, developed as part of the 2022 offering of EE194/EE290C Special Topics in Circuit Design class.

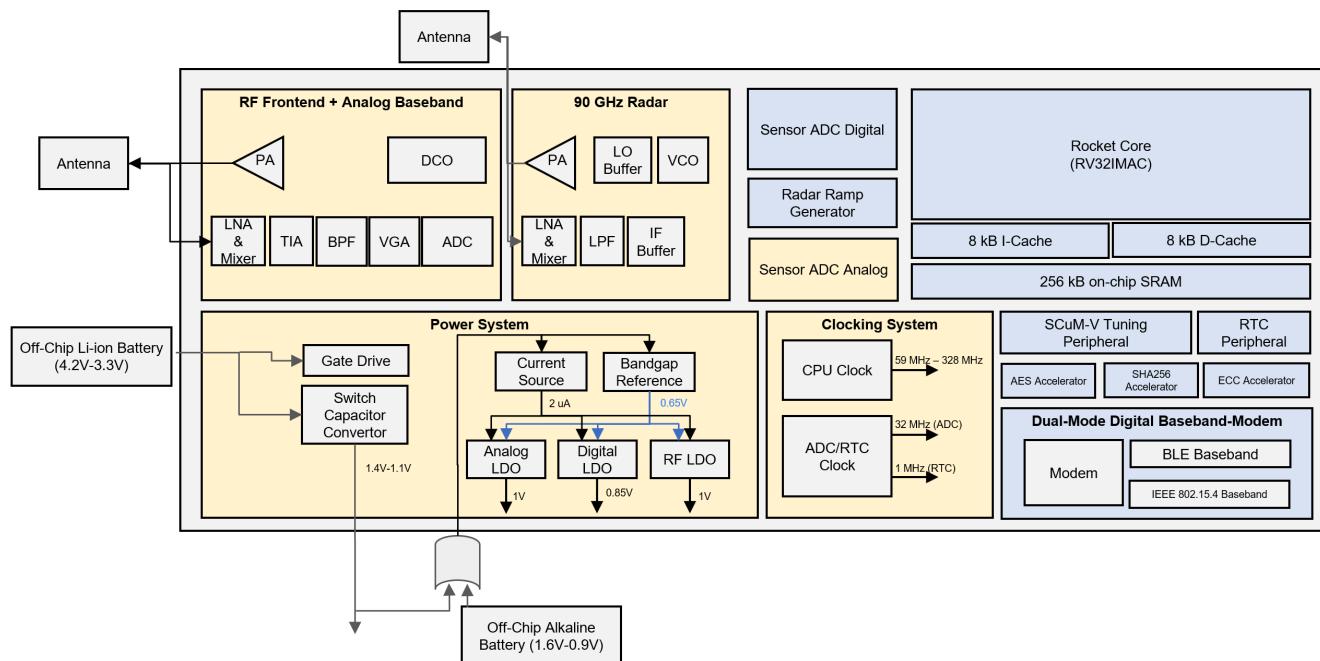
The SoC is designed around the RISC-V 5-stage in-order Rocket processor, generated by the RocketChip generator, embedded in the Chipyard framework. It features a crystal-free design with custom dual-mode IEEE 802.15.4 and Bluetooth Low Energy (BLE) transceiver. The SoC is implemented in the Intel 16 FinFET process.

The SCuM-V chip offers several GPIO pins, a single UART bus, QSPI Flash, and an on-chip, crystal-free RTC with capture and compare registers. This iteration of the chip did not support I2C or an AES accelerator.

SCuM-V23 has several new features introduced over its predecessor SCuM-V22:

- 32-bit Rocket core with cryptography acceleration
- Switched capacitor DC-DC converter cascaded with LDOs
- 90 GHz FMCW radar transmitter
- 18-bit μ V-precision ADC

2.1. Block Level Diagram + Chip Features



2.2. 32-bit RISC-V Core

The SoC includes a 32-bit Rocket, which has a high-performance single-issue in-order execution pipeline, with a peak sustainable execution rate of one instruction per clock cycle. The core supports Machine and User privilege modes as well as standard Integer, Multiply, Atomic, Floating-Point and Compressed RISC-V extensions (RV32IMAFc).

Detailed description of the core is in Chapter 3.

2.3. On-Chip Memory System

The Rocket core has a 8kB I\$ and D\$ and a 256kB scratchpad acting as L2 memory.

2.4. Busses

The design includes a system, peripheral, control, and front bus. The Rocket Tile communicates through the system bus to the remaining three buses. The control bus accesses BootROM, PLIC/CLINT, and the debug interface. The frontend bus handles DMA devices.

2.5. Interrupts

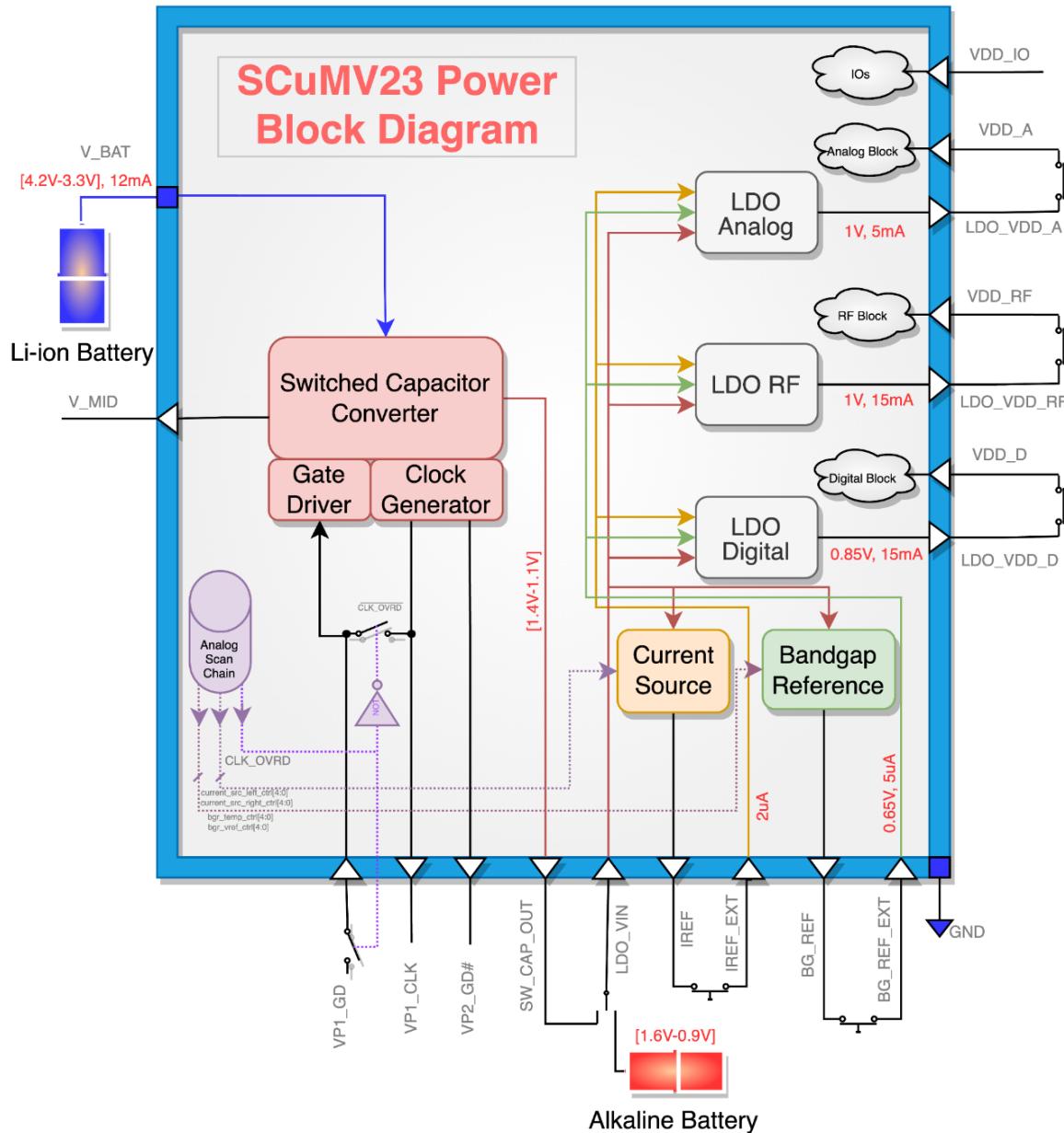
Two interrupt controllers for handling MMIO are attached to the control bus, the platform level interrupt controller and the core local interrupts. Core local interrupts are used for timer and the software while the PLIC interacts with the peripherals.

2.6. Dual-Mode BLE/IEEE 802.15.4 Transceiver

A dual-mode transceiver that supports both Bluetooth Low-Energy and IEEE 802.15.4 modes of operation. In the Bluetooth LE operating mode, the transceiver is capable of sending and receiving BLE Link Layer packets at a rate of 1Msym/s. In IEEE 802.15.4 operating mode, the transceiver sends and receives 802.15.4 PHY packets at 2 Msym/s.

The CPU is able to provide packet payloads to the digital baseband using DMA. Incoming packets will trigger multiple interrupts and provide the packet to the CPU via DMA. Control, status, and tuning registers are exposed to the CPU via MMIO. This includes a large suite of parameters of the transceiver (e.g. radio mode, channel tuning) that are accessed with these memory-mapped registers.

3. Power System



SCuM-V23 has 3 power domains:

1. VDD_A - 1V, 5mA expected
2. VDD_RF - 1V, 15mA expected
3. VDD_D - 0.85V, 15mA expected

3.1. Switched Capacitor Converter

Ladder 3:1 Topology

- All capacitors & switches are rated at V_{out} .
- Externally Reconfigurable for 3:1, 2:1, 3:2 for exploiting full battery voltage range.

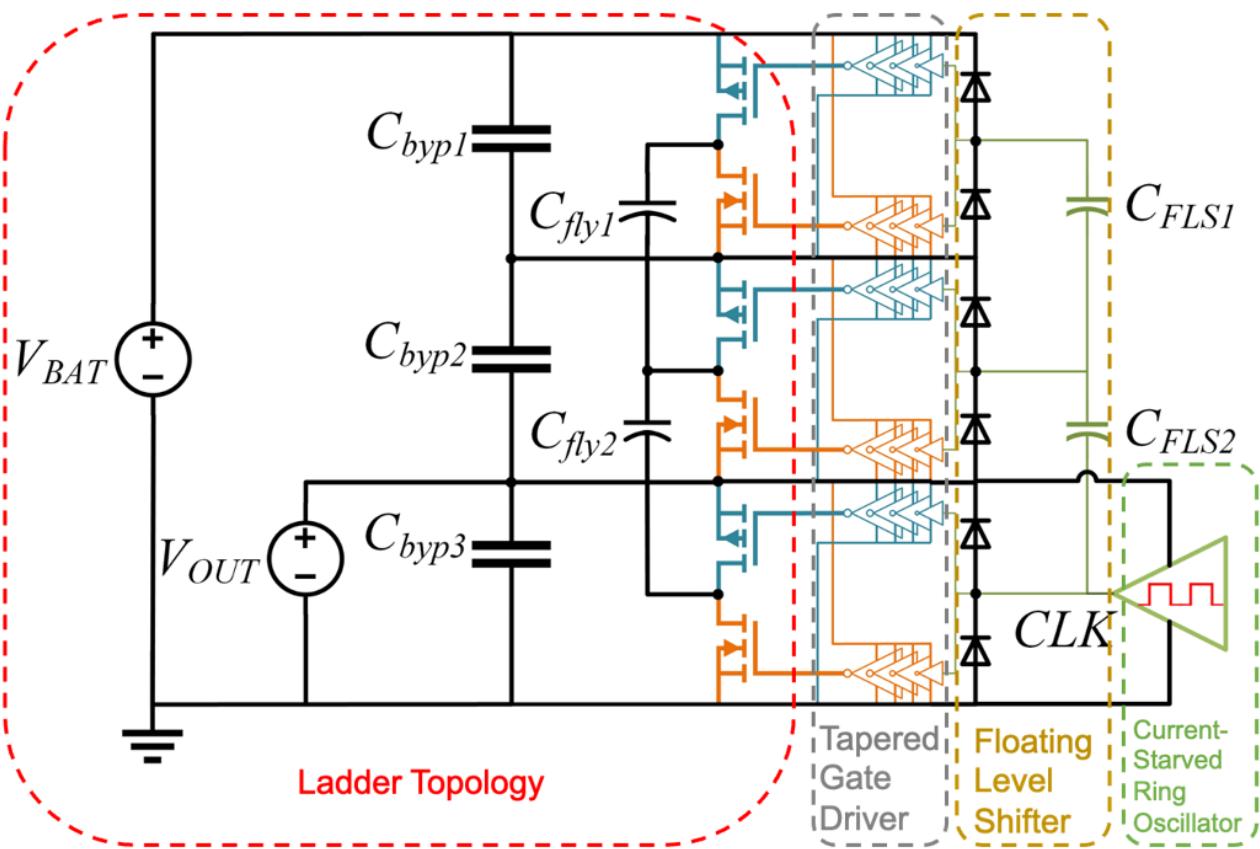


Figure 1. Switched capacitor converter schematic

Simulated Efficiency Variation with Load

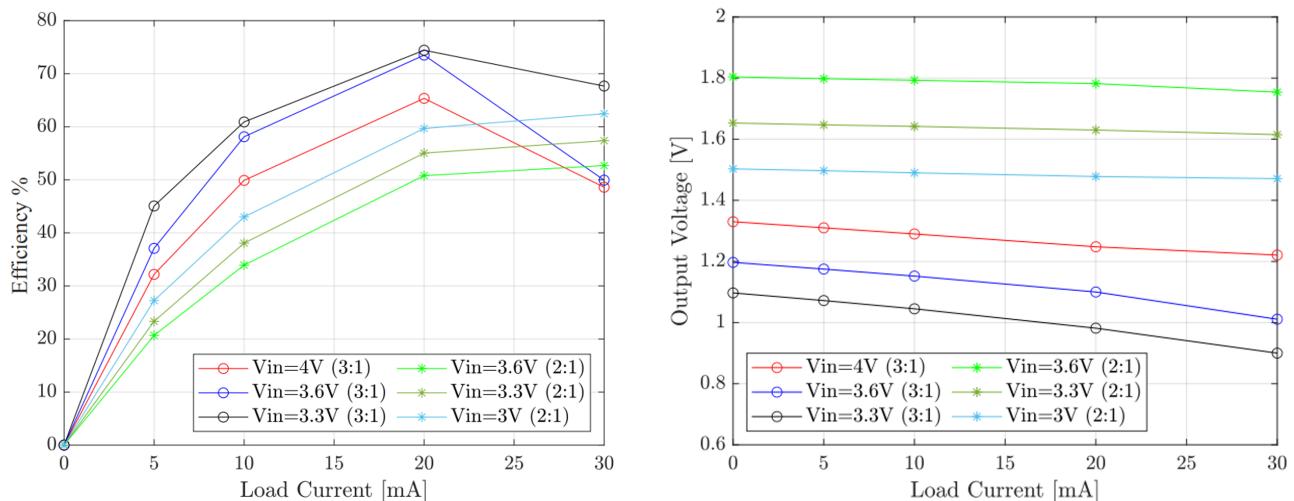


Figure 2. (Left) Efficiency Variation with Load. (Right) Vout Variation with Load

Table 1. Simulated performance

Simulated Performance Parameters	Details
Clock Frequency	85MHz-120MHz
Peak Power Efficiency	75% (Pre-layout)
Full-load Efficiency	68% (Pre-layout) and <50% (Post-layout)
Max Vout_ripple (pk-pk)	50 mVpp
Power Density at full-load	65mW/mm ² @ 100MHz

Table 2. Overview of High-Level Power Converter Design

Parameters	Details
Total Capacitance	6.5nF
Capacitor Type	MOM+MOS
Vin Range	4.2 V-3V
Vout Range	1.8V-1.1V
Conversion Ratios	3:1, 2:1
Load Range	5mA to 30mA
Switched-Capacitor Area	450,000 um ² (75% of Power Block)
Max Output Power	30 mW

3.2. Low Dropout (LDO) Voltage Regulators

The design of the LDO is a 2-stage amplifier where the output of the differential amplifier drives the gate of the PMOS at the second stage and the output of the drain of this PMOS is the regulated voltage. This voltage is then passed through a resistor-feedback network to get the desired regulated voltage for different power domains. An additional capacitor is attached at the output voltage as well to stabilize the LDO. Since each domain requires different voltages, the resistors values were chosen to such that the feedback yielded the desired voltage for different domains (1V for the RF analog, and 0.85V for Digital).

Table 3. Simulated performance

	LDO_RF (1 V)	LDO_Digital (0.85 V)	LDO_Analog (1 V)
Vin	1.8 V - 1.2 V	1.8 V - 1.2 V	1.8 V - 1.2 V
Vout range (at full load)	1.08 V - 0.81 V	0.83 V - 0.73 V	1.19 V - 0.978 V
Max Output Power	15 mW	15 mW	5 mW

3.3. Current Reference

Distributes 2 uA current reference to other blocks, includes a start-up circuit.

Parasitic extraction results:

- Typical conditions yields reference current ranging from 1.98 uA to 2.4 uA (~20% variation)
- Large variation with corners
- Worst case: ff_hot at 3.04 uA

3.4. Bandgap Reference

The Bandgap Reference Circuit (BGR) can be splitted into 3 parts: startup circuit, 2-stage op-amp, and the core BGR circuit. There are 2 included RDACs for the temperature coefficient and output voltage (Vref) adjustment.

Table 4. Key parameters and simulated performance

Parameter	Spec
Output, VBAT=1.6V	~650 mV
Power Consumption, T=27C	~1.8 mW
PSRR, f=1 MHz	~58 dB
Layout Area	~ 7,500 um ²

3.4.1. Core BGR Circuit

The temperature compensation coefficient is determined by the ratio of R3/R2, while the output voltage level is determined by the ratio of R4/R2, assuming R1=R2. R3 and R4 are eventually implemented as RDACS to enable tuning of temperature coefficient and output voltage shifting for LDO Vref supply.

3.4.2. 2-Stage Op-Amp

A two stage opamp with Miller compensation and pole cancellation was used.

3.5. Utility Current DAC

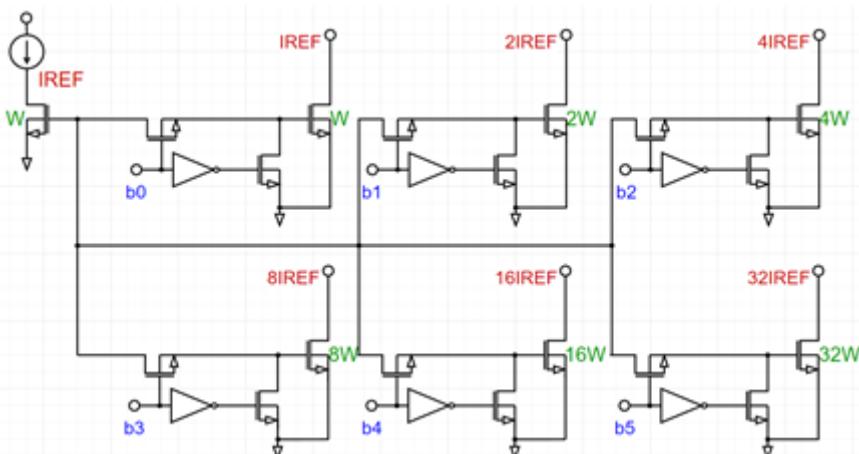


Figure 3. Utility Current DAC schematic

Table 5. Key parameters and simulated performance

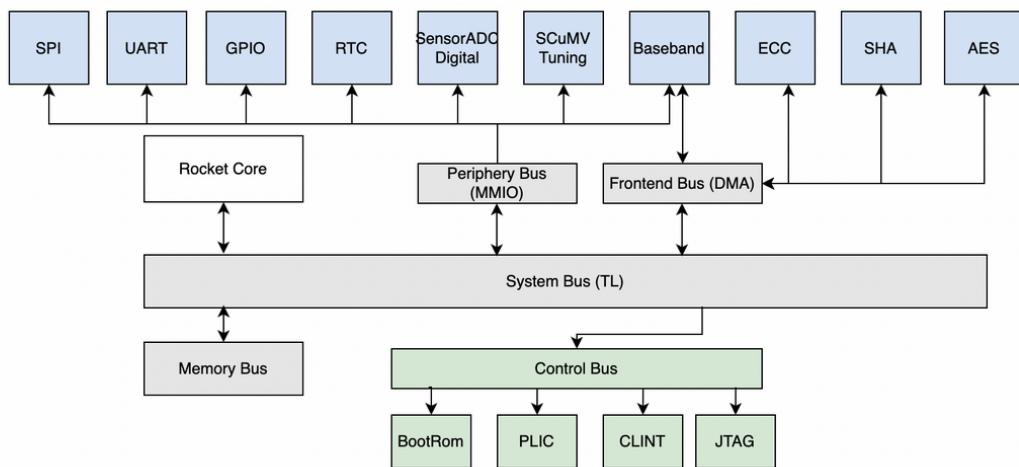
Parameter	Spec
Output @ 1.6 V	Up to $63 * I_{REF}$
Power @ 126 uA	$\sim 50 \mu\text{W}$
Input Current Range	[0.1 uA, 20 uA]
Load Voltage	$\sim 150 \text{ mV}$
Average DNL Error	1.73%
Overall INL Error	$\sim 2 \mu\text{A}$
Estimated Area	800 μm^2

- 6-bit binary weighted current mirror scaling
- Uses I_{REF} input as LSB

4. Oscillators

- What clock domains are present?
- How is the system intended to work if fully functional?
- What tuning accessibility do we have?
- What are minimum/maximum frequencies?

5. Digital Core



5.1. Key Features

- RV32IMAF Single Rocket Core
- 3 clocks for CPU core and peripherals:
 - 200 MHz CPU clock
 - 32 MHz ADC clock for BLE/IEEE 802.15.4 baseband-modem and precision sigma-delta ADC
 - 1MHz RTC clock
- On-chip SRAM
 - 256kB scratchpad ("L2")
 - 8kB 2 way I\$ & D\$
- JTAG, UART, QSPI Flash, GPIO
- Interrupt controller
 - PLIC (platform level)
 - CLINT (core level)
- RTC capture and compare peripheral

5.2. Power Consumption

Table 6. Power consumption comparison of the core running a program with baseband register read/write and ALU usage. VCC=0.85V, CPU clock=200MHz

	Internal Power (mW)	Switching Power (mW)	Leakage Power (mW)	Total Power (mW)
SCuM-V22 in Lab	11.9		36	42.5
SCuM-V23 dynamic power simulation	7.9	3.2	0.060	11.2

Based on simulations, total power consumption of the core decreased by 73% from SCuM-V22 to SCuM-V23. This improvement was achieved by reducing leakage power in the SRAM cells.

5.3. Boot Process

Method	BOOT_SEL Value	Boot Address	Interface	Boot Process Description
Self Boot	1	0x2000_0000	SPI	SPI flash content is copied to memory Jump to self boot base address
Tethered Boot	0	0x0000_0000	TSI & JTAG	<p>1. Boot ROM configures a trap handler, and enters a wait for interrupt loop.</p> <p>2. Use JTAG, TSI, or other external debugging tools to program the on-chip memory</p> <p>3. Use JTAG, TSI, or other external tools to trigger a software interrupt (MSIP)</p> <p>4. Boot ROM handles interrupt, jumps to the program memory.</p>

- Core architecture
- How much memory
- How do we boot
- Where do we find simulations/programs that were run previously?
- Where is the toolchain?
- Is this the right memory map?

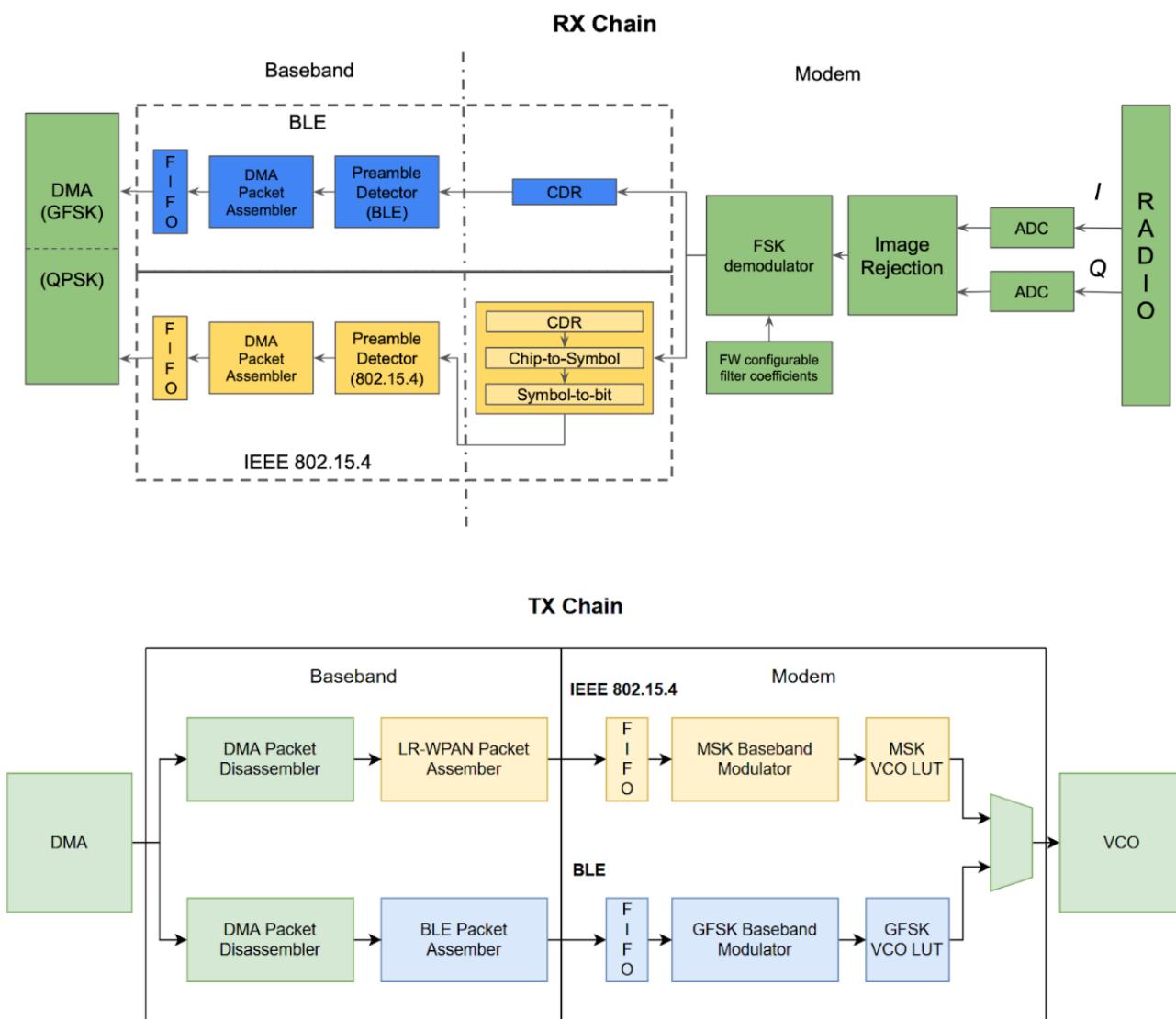
6. Digital Baseband-Modem

6.1. 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.

6.1.1. Architecture Overview



6.1.2. Baseband-Modem Frontend

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
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

ADDR	DATA	Size (bits)	Description
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)
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

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 0xFFFFFFFF6 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

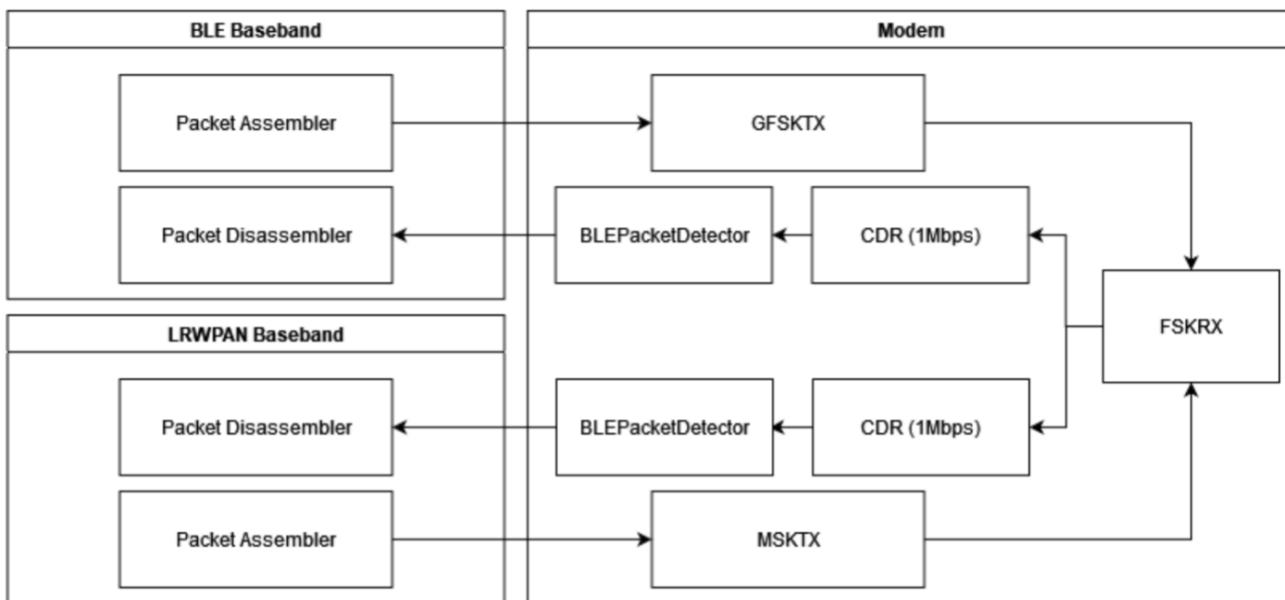
Turns on both the TX and RX paths according to the specified loopback mask before passing the specified number of PDU bytes in a loop. For simplicity the return data is stored at <load address + total bytes> rounded to the next byte aligned address. The loopback mask is used to determine at which point 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

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



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.

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.

6.1.3. 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 7. 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.

6.2. IEEE 802.15.4 LR-WPAN Baseband

Baseband 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.

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.

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

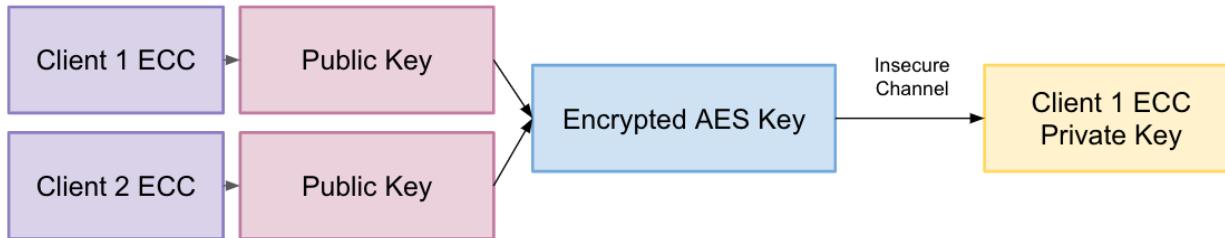
The preamble for 15.4 is a 32-bit sequence of all zeros. The SFD is loosely analogous to the Access-Address (AA) of the BLE Baseband in that they both uniquely identify the start of the packet for the protocol used.

6.3. Bluetooth Low Energy (BLE) Baseband

7. Encryption Modules

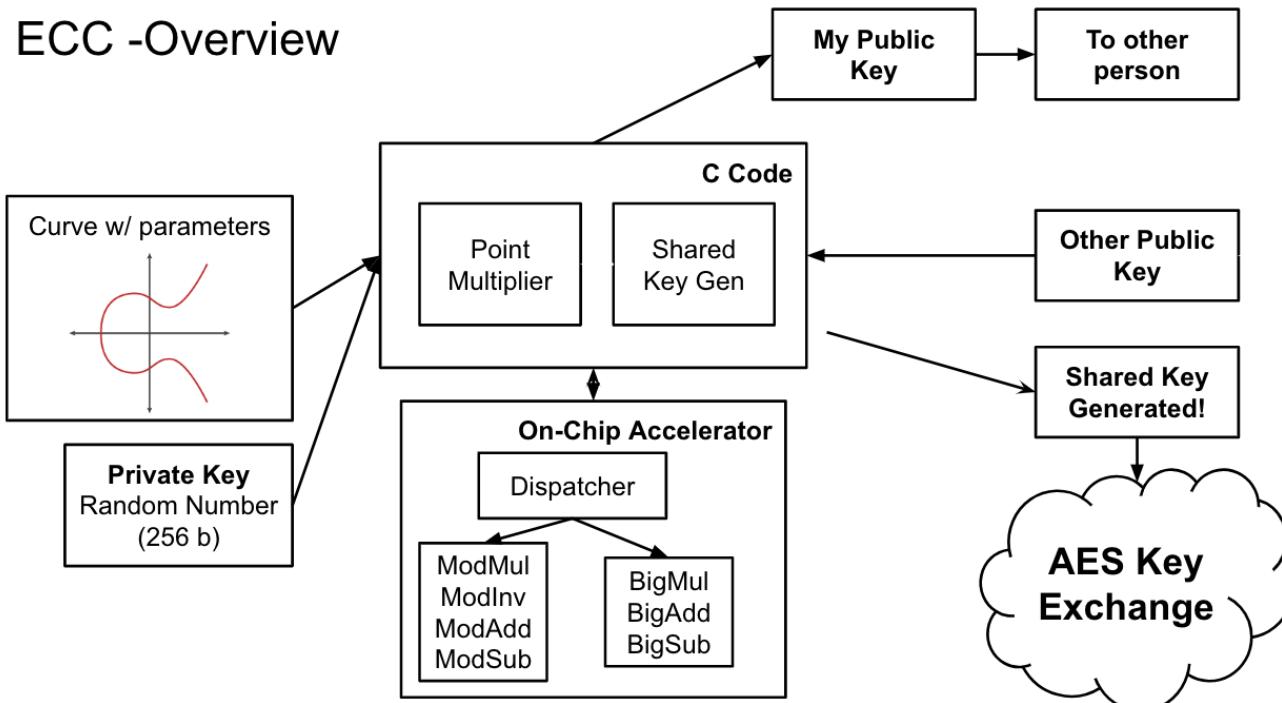
Establish secure communication between two devices in following steps:

1. Share key Elliptic Curve Cryptography (ECC)
2. Sender encrypts message using Advanced Encryption Standard (AES) and shared key
3. Receiver decrypts message using AES and shared key
4. Receiver verifies message integrity using SHA-256



7.1. ECC

ECC uses Chisel implementation and calls the Accelerator through C for math operations. Multiplier sizing is optimized.

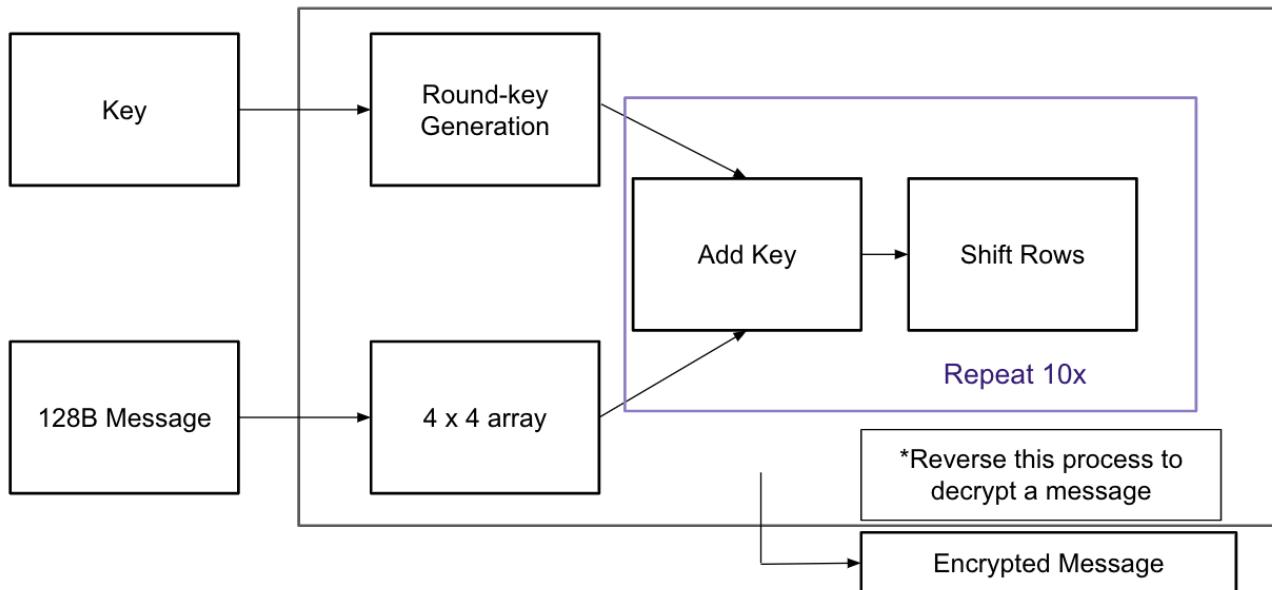


7.2. AES

AES uses Chisel implementation and AES is synthesizable on a Rocket chip. Users can call AES accelerator functions through software by utilizing software interrupts. The process involves loading the encryption key and message, initiating the encryption or decryption process, and optionally utilizing diagnostic tools to monitor cycle time for computation completion. The implementation has been thoroughly tested using C testbenches and validated with third-party online software tools. On average, it takes approximately 6 cycles per word, which

equates to about 10 nanoseconds per byte for AES operations.

AES - High Level Overview



7.3. SHA-256

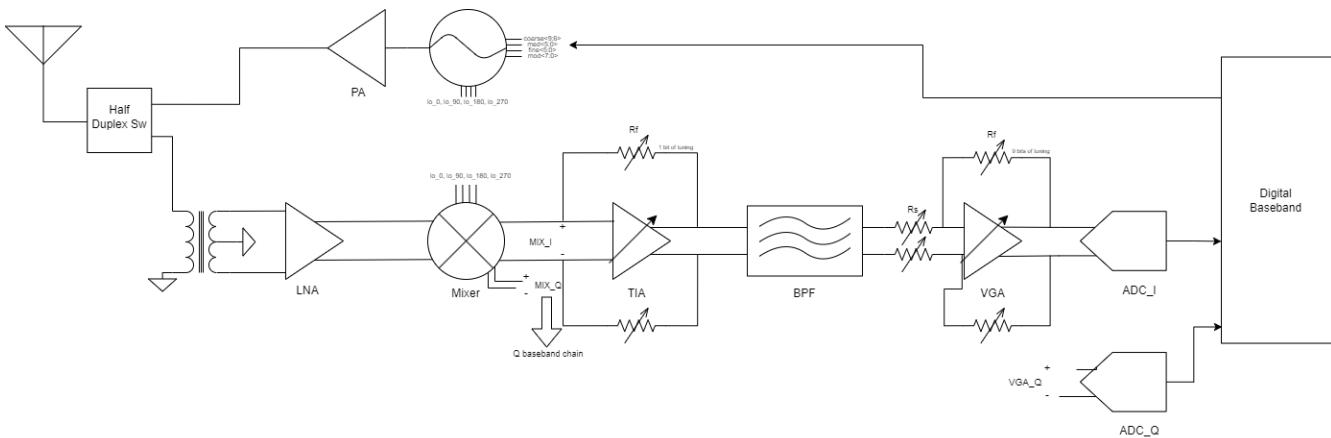
SHA-256 part combines multiple hash functions for its operation. It divides data into eight 16-bit chunks, denoted as A to H, and performs bitwise XOR, AND, and bit-shifting operations on these chunks. It shares similar implementation and capabilities with an AES accelerator, with the exception of its hashing functionality. The project includes C testbenches, similar to those used for AES, and has been verified using third-party online software tools. On average, it takes approximately 107 cycles to process 256 bits of data, which translates to about 70 nanoseconds per byte for hash computations.

8. 2.4 GHz Tranceiver

8.1. Radio Architecture

The radio operates within the frequency range 2.4 to 2.4835 GHz covering 70 1MHz RF channels. The receive chain should be sensitive to an input power range of -70 dBm to -10 dBm and achieve a BER of 0.1% or an SNR of 12.5 dB.

8.2. Radio Final Block Diagram



8.3. All Interface Pins

Block	Analog/Signal Pins	Digital Pins
LNA, Mixer	rx, Ina_p, Ina_n, iref_Ina*, Io_0, Io_90, Io_180, Io_270	N/A
TIA	mix_i_p, mix_i_n, mix_q_p, mix_q_n, tia_i_p, tia_i_n, tia_q_p, tia_q_n, MUXED_ANALOG_IN_P, MUXED_ANALOG_IN_N, MUXED_ANALOG_OUT_P, MUXED_ANALOG_OUT_N,	en_tia_i, en_tia_q, vga_gain_ctrl_i<9>, vga_can_ctrl_q<9>
BPF	tia_i_p, tia_i_n, tia_q_p, tia_q_n, iref_bpf_i_1, iref_bpf_i_2, iref_bpf_q_1, iref_bpf_q_2, MUXED_ANALOG_IN_P, MUXED_ANALOG_IN_N, MUXED_ANALOG_OUT_P, MUXED_ANALOG_OUT_N	en_bpf_i, en_bpf_q, bpf_q_clp_0<3:0>, bpf_q_clp_1<3:0>, bpf_q_clp_2<3:0>, bpf_q_chp_0<3:0>, bpf_q_chp_1<3:0>, bpf_q_chp_2<3:0>, bpf_q_chp_3<3:0>, bpf_q_chp_4<3:0>, bpf_q_chp_5<3:0>, bpf_i_clp_0<3:0>, bpf_i_clp_1<3:0>, bpf_i_clp_2<3:0>, bpf_i_chp_0<3:0>, bpf_i_chp_1<3:0>, bpf_i_chp_2<3:0>, bpf_i_chp_3<3:0>, bpf_i_chp_4<3:0>, bpf_i_chp_5<3:0>
VGA	iref_vga_i, iref_vga_q, vga_i_p, vga_i_n, vga_q_p, vga_q_n, MUXED_ANALOG_IN_P, MUXED_ANALOG_IN_N, MUXED_ANALOG_OUT_P, MUXED_ANALOG_OUT_N	en_vga_i, en_vga_q, vga_gain_ctrl_q<8:0>, vga_gain_ctrl_i<8:0>
VCO	LOp, LOn, iref_vco, Io_0, Io_90, Io_180, Io_270, tx_p	en_vco_lo, vco_freq_reset, vco_cap_mod<0:7>, vco_cap_med<0:5>, vco_cap_coarse<0:9>

*iref_Ina is, unfortunately, routed quickly from iref_tia_i, but provides the same current

Top-Level Layout Floorplan TODO: grab screenshot of final layout and identify important blocks

8.4. Transmitter

The transmitter is designed to operate in the 2.4 GHz ISM band and be standards-compliant with IEEE 802.15.4 and BLE. The tuning range of the 4.8 GHz DCO is 4.88 GHz \pm 0.5 GHz with <40 ppm of resolution (<192 kHz), corresponding to 2.44 GHz \pm 0.25 GHz after the divider.

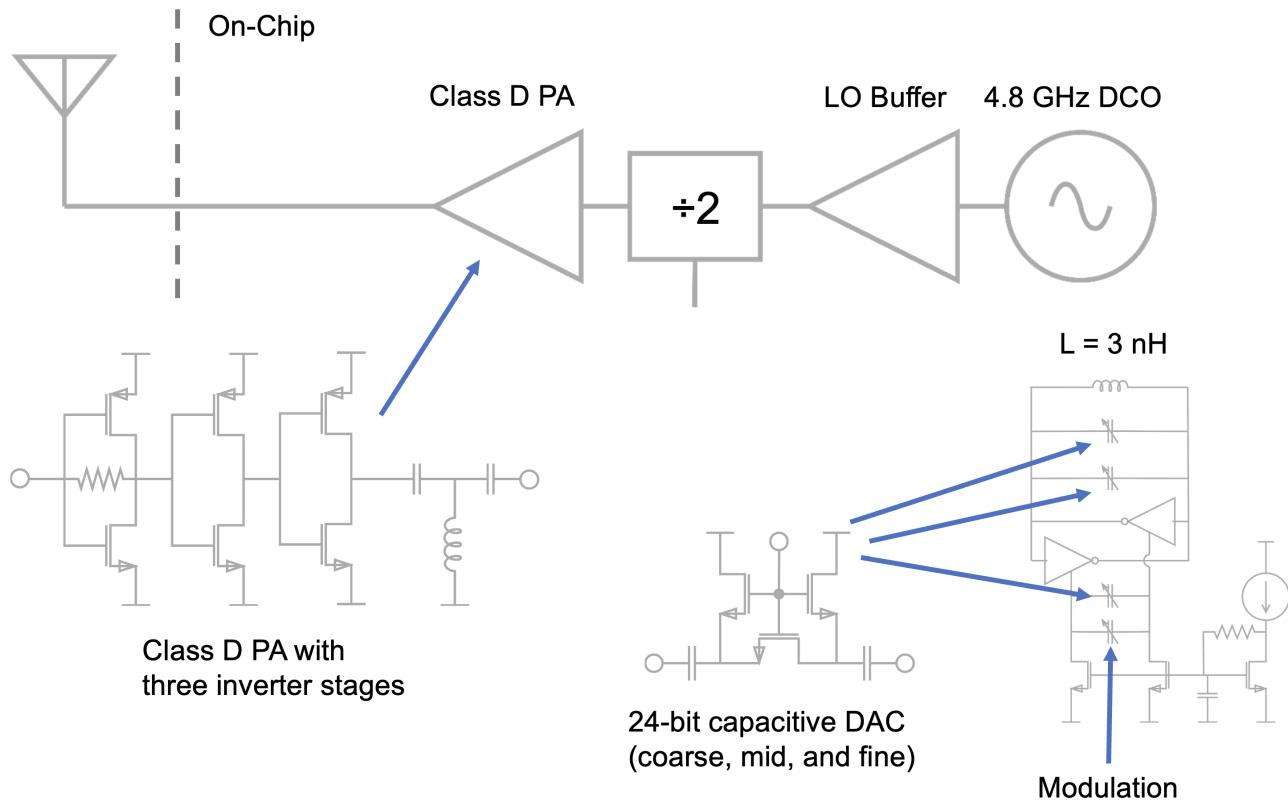


Figure 4. Transmitter schematic. The divider divides the 4.8 GHz DCO signal down to 2.4 GHz and also generates the differential in-phase and quadrature components for the mixer.

8.4.1. DCO Performance

Referenced to 4.8 GHz, the following table displays the specifications and the achieved performance with parasitic extraction at the typical corner.

Specifications	Desired	Simulated
Power consumption [mW]	<1 mW	0.871 mW
Tuning bandwidth	20% of center frequency (4.88 GHz)	18.12% (4.8 GHz center)
Coarse step resolution	<64 MHz (1024 MHz bandwidth)	50.2 MHz (803 MHz bandwidth)
Medium step resolution	<4 MHz (180 MHz bandwidth)	2.31 MHz (74 MHz bandwidth)
Fine step resolution	<192 kHz (10 MHz bandwidth)	67.2 kHz (4.3 MHz bandwidth) Modulation step resolution

8.4.2. PA Performance

The PA is a class D PA consisting of three inverter stages. In simulation across corners, it outputs between 2.2 dBm and 3 dBm with an efficiency between 33% and 35%.

8.5. Design Flow, Tradeoffs, and Decisions

8.5.1. RF Frontend

The RF receive chain consists of an input balun that converts the single ended antenna signal to a differential input signal and provides help with impedance matching. Following the balun is a fully differential low noise amplifier with inductive degeneration that produces an RF current. The RF current is mixed, in current mode, by a 4-phase differential passive mixer which is then converted to an IF voltage by the TIA. For designing these circuits taking EE 142 is a must, and solid understanding of electromagnetics is very useful.

Balun

LNA

To be filled

Mixer

The mixer is a 4-phase differential passive mixer. Review 142/242B notes for analysis and implementation. Reused from SCuM-V22, designed by Rami Hijab.

TIA*

The TIA converts the mixer output current to a baseband output voltage. Due to time constraints, we re-used the op-amp from the baseband chain for this TIA. However this should have been its only resistively biased inverter (or other topology) for better noise performance. The key to the TIA (what was different from the baseband chain) was the resistive feedback. This was done last minute to meet the low and high gain requirements. The addition to make is to make sure that the system is invariant to PVT variation. This can be done by making the current source of the LNA a $1/R$ reference so that the LNA G_m cancels with the R in feedback with the TIA.

8.5.2. Baseband

The baseband chain consists of many individual blocks, and this means that not too much time could be spent on optimizing designs for each one. Building off of Neelesh's experiences in the Spring 2022 iteration of the class, the main goals were to decrease power consumption and area of the baseband by optimizing the BPF implementation.

VGA

Stage 1

High gain folded cascode stage. Chosen to more independently control the stage 1 gm and the stage 1 R_{out} . Headroom limitations on the long-channel devices makes telescopic hard to achieve. Input pair is very large to minimize DC offset voltage (proportional to area of the input devices). Quite large intrinsic gain. Initially I wanted to support 60dB of dynamic range on the CL performance of the op-amp, which required 90dB or so of open-loop gain at the minimum. We ended up using 2 gain stages (TIA = stage 1, VGA = stage 2) so the second VGA only needed a max of 30dB dynamic gain. So, this stage can be optimized more for the new spec. BW concern is not that large since the IF of the baseband is only 2MHz. Design approach: I used the topology of the previous semester as a starting point before beginning to test various currents, widths, device flavors, etc. I believe that 240B HW3 of Sp22 is the kind of assignment (as Prof. Niknejad for someone's good solutions to this, or his own, or email me for mine, etc. etc.) that teaches you how to design the main baseband op-amp block (stage 1 at last), and is a really good learning opportunity for anyone interested in this work. When it comes down to testing, you have to know why the amp behaves the way it does, and whether it does what you expect (voltage biasing, currents, VGS/VDS, etc.). The difference between systematic design and just getting something that happens to work (honestly, I was stuck in this regime for a while) is that if a testbench result is puzzling, you should have an idea of how to proceed. Of course, message on slack/Piazza/ bring it up during weekly meetings, but individual progress unhampered by others is critical to get things done on time.

Stage 2

Common-source class A stage. Power was not a concern for the spec of this class, and so other topologies were not considered (the CS amp stage is quite simple to bias and understand as well). Spent enough time to make sure this block was biased correctly and could support a large enough capacitive load. In reality, this varies; for the TIA, this cap load is the input of the BPF, which isn't very large. But the output of the second VGA is the ADC input, which is a much larger (~pF) cap. Miller Compensation sets some limits on the relation between input/output stage gm values and the cap load you can drive (240B slides).

CMFB Circuit

Compares the output common-mode voltage to the desired level to set the top PMOS gate voltage. Does not involve sensing resistors (minimizes area). Fully-differential, PMOS diode loads. In hindsight, the second CMFB loop may not be needed at all (also helps stability to only have 1 CMFB loop). The PMOS bias can be a voltage source (add a branch to the biasing network).

Biasing

Functional but can be optimized. The structure can be made easier to control via more independent branches (more power but again, not a concern).

BPF

This block was not hard to design given a good op-amp; there are many papers and sample designs available for a multiple-feedback active band-pass filter composed of individual HP/LP stages, along with design equations. In this case, a high pass filter is followed by a low pass filter to create the bandpass transfer function.

A redesign of the op amp used was done to reduce area, power consumption.

A 5T configuration was used.

A current mirror set up was used, with resistive CMFB using the four resistors at the bottom. Miller compensation is also implemented to ensure stability between the fully differential outputs as denoted by v_{outp} , v_{outn} .

Upon simulation of the extracted view, it was found that the pass band of the BPF had shifted as the previous RC values for the high pass and low pass stages were not tuned properly. However, current simulations of the taped-out design show the passband near 12 MHz, and needs a fix.

ADC

This is a reused block from SCuM-V22 and developed by Zhaokai Liu.

A pseudo-differential input ADC, one can specify a high and low reference voltage in which comparison should be done. By taking the lower reference voltage as "ground", and the pseudo differential input signal as the input.

8.6. Simulation

Refer to the "Analog RF Simulations" link in the Resource Index for information about locations of testbenches and final tapeout designs.

9. 94 GHz FMCW Radar Transmitter

The frequency-modulated continuous wave (FMCW) radar transmitter is designed to generate linear chirps with the following high-level specifications:

Parameter	Value
Center frequency	94 GHz
Maximum chirp bandwidth	1 GHz

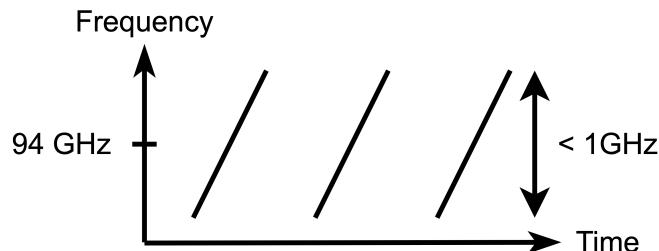


Figure 5. Linear chirps over time with a center frequency of 94 GHz.

The radar transmitter is implemented with a Class A voltage-controlled 47 GHz oscillator with a varactor driven by a ramp voltage. The ramp voltage is generated digitally by an 8-bit counter register whose output feeds into an R-2R DAC. The output of the VCO is fed into a frequency doubler, and the resulting 94 GHz signal is amplified by the PA.

A center frequency of 94 GHz was chosen because Yahia, a graduate student advised by Prof. Niknejad, had recently taped out a 90 GHz PA in February 2023 and Prof. Niknejad suggested to use Yahia's PA for this radar transmitter. As a result, the PA is courtesy of Yahia.

9.1. System Overview

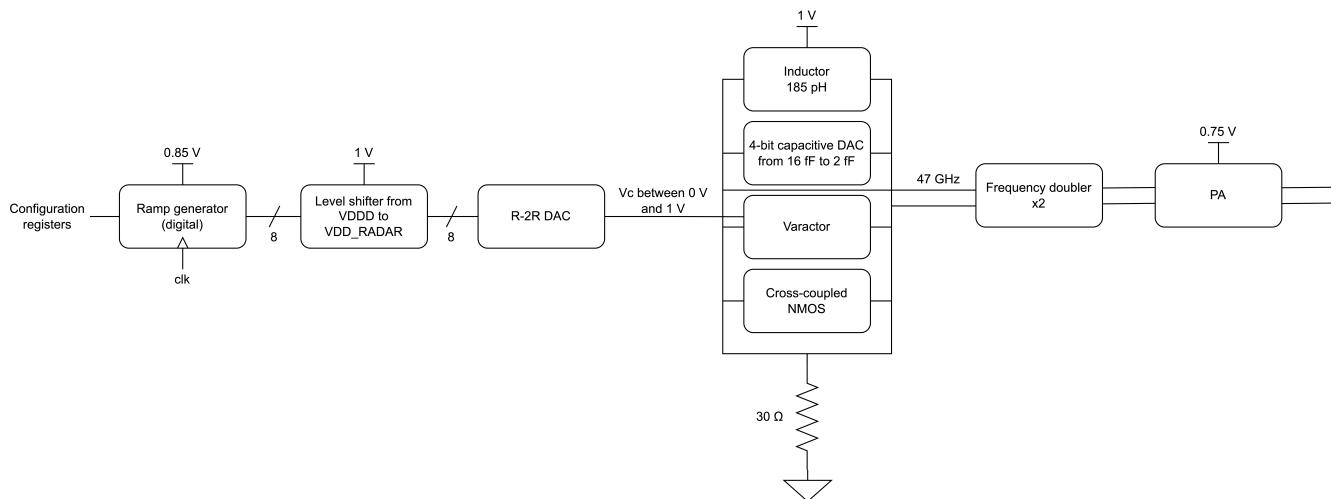


Figure 6. Radar system-level diagram.

Unlike the other blocks on the chip, the radar transmitter has two dedicated power rails, **VDD_RADAR** and **VDD_PA**, that are not regulated internally and must be sourced externally. Nominally, **VDD_RADAR** should be 1 V, and **VDD_PA** should be 0.75 V.

9.2. Ramp Generator

The ramp generator is a digital circuit responsible for outputting an 8-bit ramp output that feeds into the R-2R DAC. Internally, there is a 32-bit counter register that is clocked by the CPU clock and used to generate the ramp

output.

There are two states of the ramp generator: `STATE_IDLE` and `STATE_RAMP`.

- In `STATE_IDLE`, the ramp generator waits for `ramp_generator_num_idle_cycles` clock cycles before it moves to the `STATE_RAMP` state.
- In `STATE_RAMP`, the ramp output is initially set to `ramp_generator_frequency_step_start`. Every `ramp_generator_num_cycles_per_frequency` clock cycles, the ramp output is incremented. Once the ramp output has been incremented `ramp_generator_num_frequency_steps` times, the ramp output is reset to `ramp_generator_frequency_step_start`, and the state machine moves back to the `STATE_IDLE` state. The ramp is now complete.

The reason for allowing the CPU to set `ramp_generator_frequency_step_start` as well as `ramp_generator_num_frequency_steps` is that the varactor's capacitance is non-linear with respect to the control voltage. Therefore, the control registers should be set, so that the ramp voltage causes the VCO to sweep linearly over the desired chirp bandwidth.

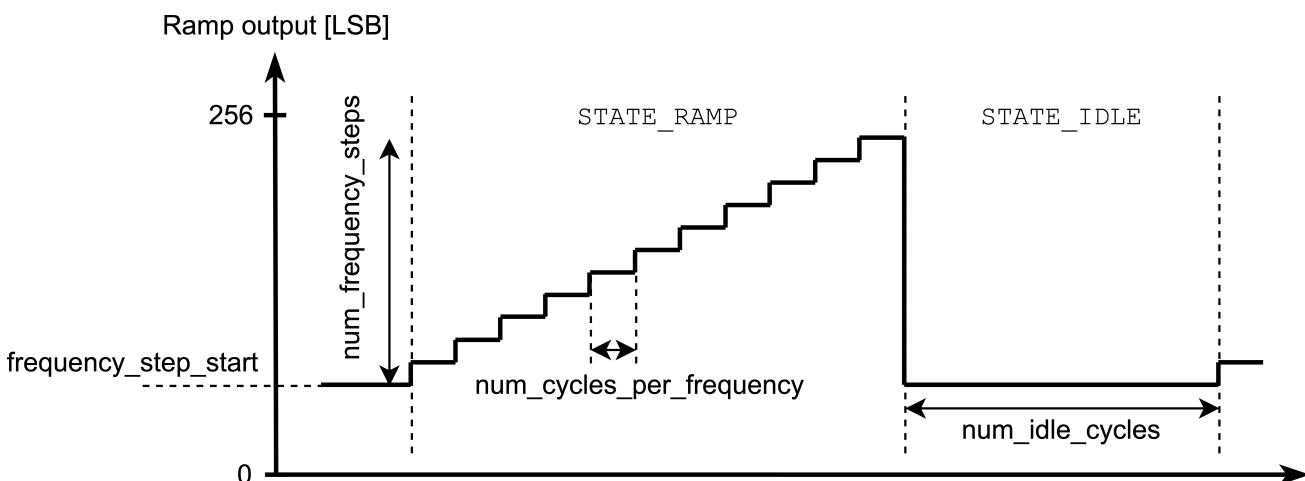


Figure 7. Ramp generator output.

9.3. R-2R DAC

The R-2R DAC converts the 8-bit ramp output from the ramp generator into an analog ramp voltage between 0 V and 1 V that controls the varactor in the VCO. The resistor values used are 1k and 2k.

9.4. VCO

The VCO is a class A VCO with a 4-bit capacitive DAC ranging from 2 fF to 16 fF. Unlike the capacitive DAC, where a higher tuning code corresponds to a lower frequency, the varactor's capacitance decreases as a function of the control voltage. However, since the frequency is sensitive to the control voltage, the control voltage is not brought out as a debug signal. In simulation, the VCO draws about 7 mW of power.

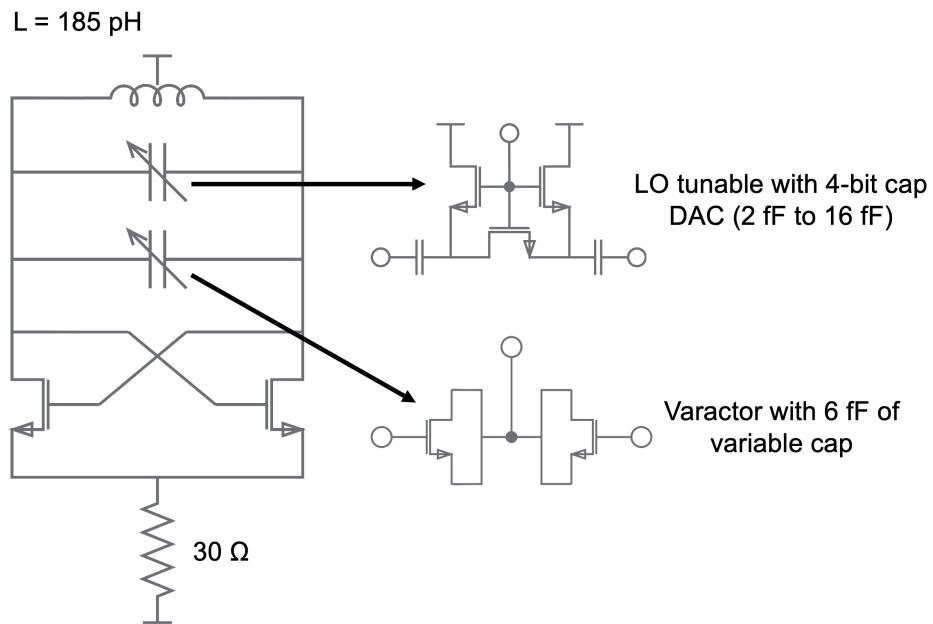


Figure 8. VCO schematic.

9.5. Frequency Doubler

The frequency doubler uses a modified cross-coupled pair with a resonant LC tank to double the VCO output's frequency. In simulation, the frequency doubler draws about 16 mW of power.

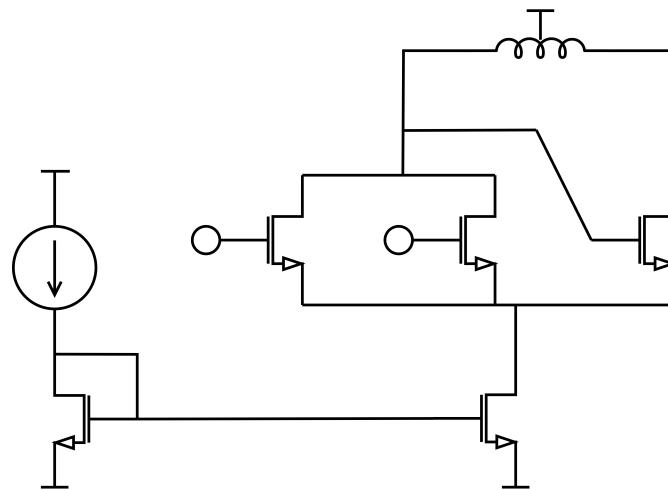


Figure 9. Frequency doubler schematic.

9.6. PA

The PA was designed and laid out by Yahia with minor modifications for SCuM-V23. The PA consists of a driver stage and an output stage and consumes up to 60 mA at a supply voltage of 0.75 V.

9.7. Register Map

The radar configuration registers are part of the **Scumvtuning** block, which contains a set of memory-mapped registers for tuning analog blocks. The base address of **Scumvtuning** is **0x0000_A000**.

Register	Name	Size (bits)	Function
0x0D	ramp_generator_clk_mux_sel	1	Unused.
0x0E	ramp_generator_enable	1	Initial 1b'1. Enables the ramp output.
0x0F	ramp_generator_frequency_step_start	8	Initial: 8b'0. Sets the initial ramp output for each ramp.
0x10	ramp_generator_num_frequency_steps	8	Initial: 8b'0. Sets the number of LSBs the ramp output should increment by for each ramp.
0x11	ramp_generator_num_cycles_per_frequency	24	Initial: 24b'0. Sets the number of clock cycles before incrementing the ramp output.
0x14	ramp_generator_num_idle_cycles	32	Initial: 32b'0. Sets the number of clock cycles to idle between ramps.
0x18	ramp_generator_RST	1	Initial: 32b'1. Resets the ramp generator's counter and ramp output.
0x19	vco_cap_tuning	5	Initial: 5b'0. Tunes the 4-bit capacitive DAC of the VCO. The MSB is unused.
0x1A	vco_enable	1	Initial: 1b'1. Unused.
0x1B	vco_div_enable	1	Initial: 1b'1. Unused.
0x1C	pa_enable	1	Initial: 1b'1. Unused.
0x1D	pa_bypass	1	Initial: 1b'0. Unused.
0x1E	pa_input_mux_sel	1	Initial: 1b'0. Unused.

9.8. Layout

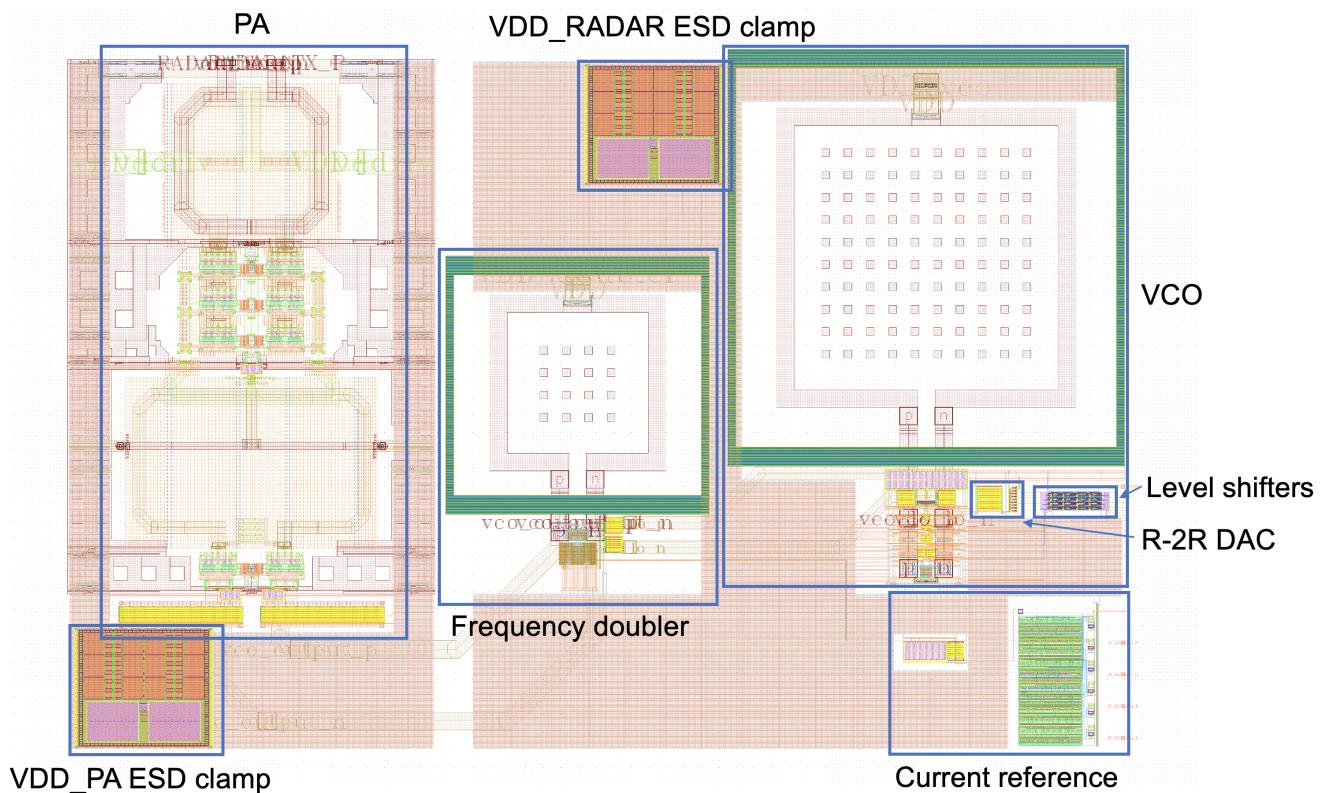


Figure 10. Final radar layout. The total area is 300 $\mu\text{m} \times 200 \mu\text{m}$.

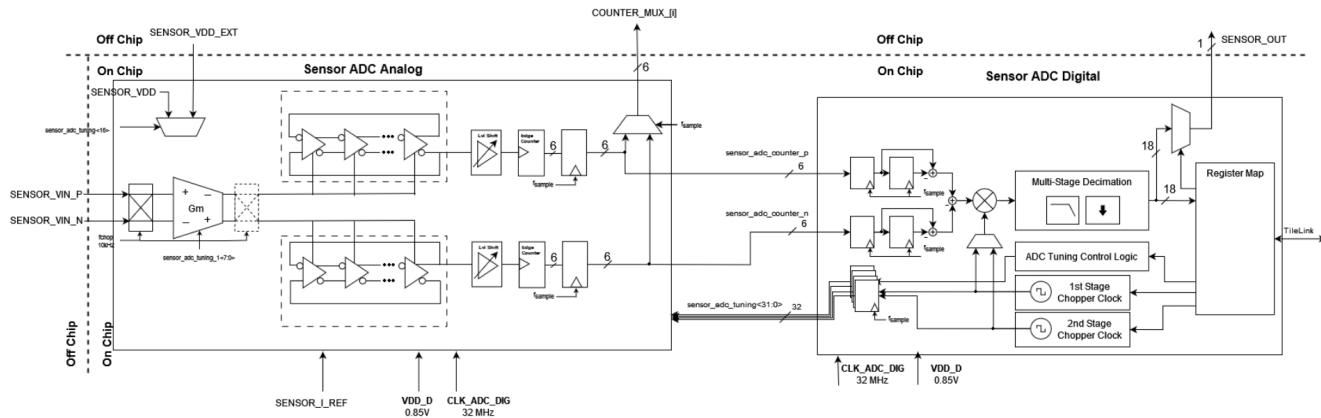
10. uV-Precision Delta-Sigma ADC

The uV-Precision ADC is designed to function as an EEG front-end with high-level specifications:

Parameter	Value
Resolution	18 bits
Noise	1.0 uVrms
Input Bandwidth	500 Hz
Spot Noise	45 nV/sqrt(Hz)

The ADC is implemented with a delta-sigma architecture, leveraging a high oversampling ratio to achieve high resolution and low noise. The input drives a current controlled oscillator, which is sampled by the on-chip 32 MHz clock to generate a digital bitstream. The bitstream is then processed by a digital decimation filter to generate the final output sample.

10.1. System-level Diagram and Description



10.2. Analog Front-End

Input impedance? Mins and maxs?

10.3. Digital Front-End

The delta-sigma ADC's DSP chain is implemented as a cascaded integrator comb (CIC) decimation filter for noise shaping. The key CIC filter parameters for this design are N=4, R=8000, M=1

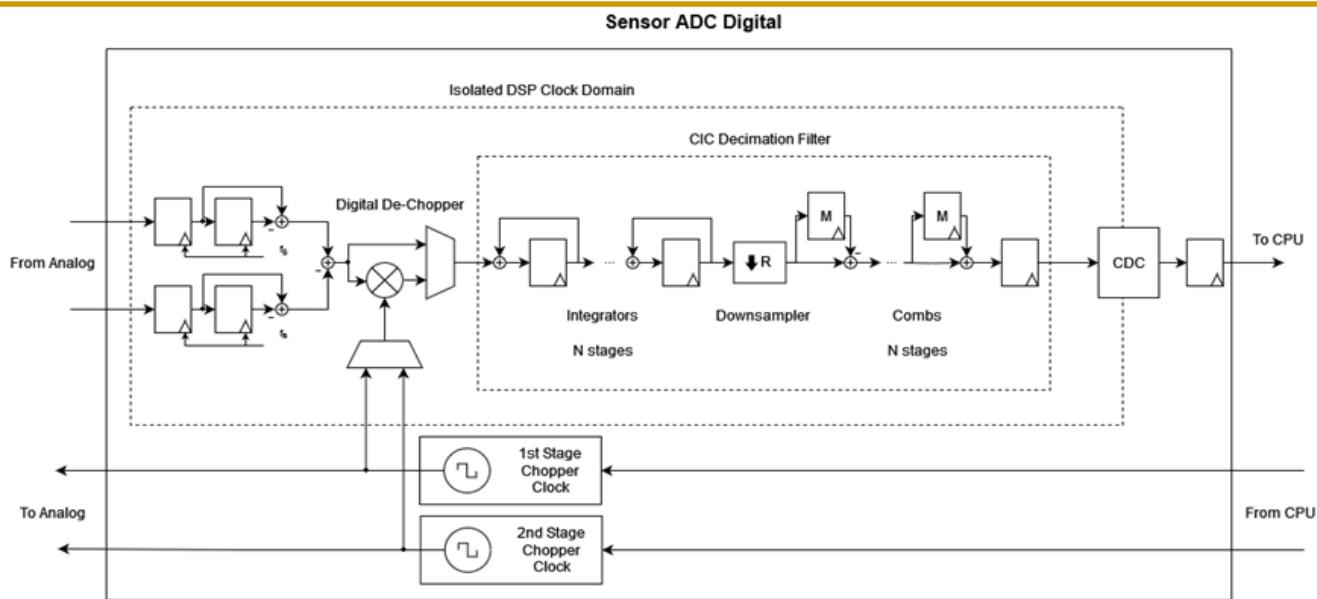


Figure 11. Digital Front-End system-level diagram

The digital frontend features self-generated, software-programmable chopper clock generators for 1st and 2nd stage choppers. A digital de-chopper/mixer with configurable clock delay to match delay of the analog front-end is also implemented.

10.4. Register Map

The base address of the sensor ADC DSP block is 0xB000. All registers in the table below are specified as offsets from the base address 0xB000.

Register	Name	Size	Function
0x00	ADC_STATUS	32	ADC_STATUS<11:6> - ADC Counter P ADC_STATUS<5:0> - ADC Counter N
0x04	ADC_DATA	20	Read ADC sample from FIFO
0x08	ADC_TUNING_0	8	ADC_TUNING_0<5:0> - Current DAC ADC_TUNING_0<6> - BIAS P + ADC_TUNING_0<7> - BIAS N
0x09	ADC_TUNING_1	8	Reserved
0x0A	ADC_TUNING_2	8	ADC_TUNING_2<0> - Sensor VDD mux select
0x0B	ADC_TUNING_3	8	ADC_TUNING_3<7:0> - Reserved
0x0C	ADC_CHOP_CLK_DIV_1	32	1st stage chopper clock divider
0x10	ADC_CHOP_CLK_DIV_2	32	2nd stage chopper clock divider
0x14	ADC_CHOP_CLK_EN	2	ADC_CHOP_CLK_EN<0> - Enable 1st stage chopper ADC_CHOP_CLK_EN<1> - Enable 2nd stage chopper
0x18	ADC_DSP_CTRL	8	ADC_DSP_CTRL<0> - Enable dechopper in DSP chain ADC_DSP_CTRL<1> - Select chopper clock used in the dechopper ADC_DSP_CTRL<5:2> - Dechopping clock delay, from 0 to 15 cycles

11. Electrical Characteristics

Include tables or descriptive text for various electrical characteristics...

12. Usage Scenarios

Describe various application scenarios, operating modes, etc.

13. Revision History

- v1.0 - Initial release
- v0.9 - Preliminary draft

14. Appendix

Additional details, references, or supplementary material...