

GC080X 应用手册

V0.4

地芯科技 & 塞防

杭州地芯科技有限公司
2023 年 2 月

目录

1、芯片简介：	3
2、原理图设计要求：	4
2.1、供电部分	5
2.2、数字部分	5
2.3、射频部分	6
2.4、时钟部分	6
2.5、其他部分	6
3、软件配置：	6
3.1、初始化和校准	6
3.2、参考时钟配置和 BBPLL	7
3.3、RFPLL 配置	7
3.4、外部本振使用配置	7
3.5、射频端口配置	7
3.6、数字接口配置	7
3.7、数字滤波器配置	10
3.8、增益控制方法	10
3.8.1、GC080x 增益分配方式	11
3.8.2、Full table 模式	11
3.8.3、Split table 模式	13
3.8.4、信号峰值和平均功率检测器	13
3.8.5、Ctrl-Out PIN	14
3.8.6、MGC 模式	15
3.8.7、Slow Attack AGC 模式	16
3.9、TX 功率控制配置	17
3.10、CTRL OUT 相关输出	18
3.11、控制输出	18
3.12、状态机	19
3.13、软复位及 QEC 配置	20
4、修订记录	22

1、芯片简介：

GC080X 是一款由杭州地芯科技有限公司开发的高性能、高集成度的超宽带 SDR 收发机。可广泛应用于几乎所有现代化数字无线通信系统。能够支持的频率范围为 30MHz 到 6GHz，可配置射频带宽能够支持小于 12KHz 到 100MHz 的范围。

该器件集成了 12bit 的模数转换器 ADC 和 12bit 的数模转换器 DAC。内置可编程模拟滤波器，支持最小 0.7MHz 带宽的模拟低通滤波器以及 TX 最大 50MHz 带宽的模拟低通滤波器，RX 最大 50MHz 的模拟低通滤波器。混频器和锁相环也都集成在芯片内部，并且发射部分集成有驱动级放大器，可以输出 8dBm 以上单音信号。

GC080X 采用直接变频架构，可实现高的调制精度和超低的噪声。该芯片具有镜像抑制校准，本振泄露校准，发射功率监控，杂散抑制以及接收通道增益校准等功能。并且包含有多芯片同步功能，适用于多芯片联合使用的场景，如 MIMO。GC080X 采用 10 mm × 10 mm、144 引脚芯片级球栅阵列封装(CSP_BGA)。

GC080X 的简单示意框图如图 1-1

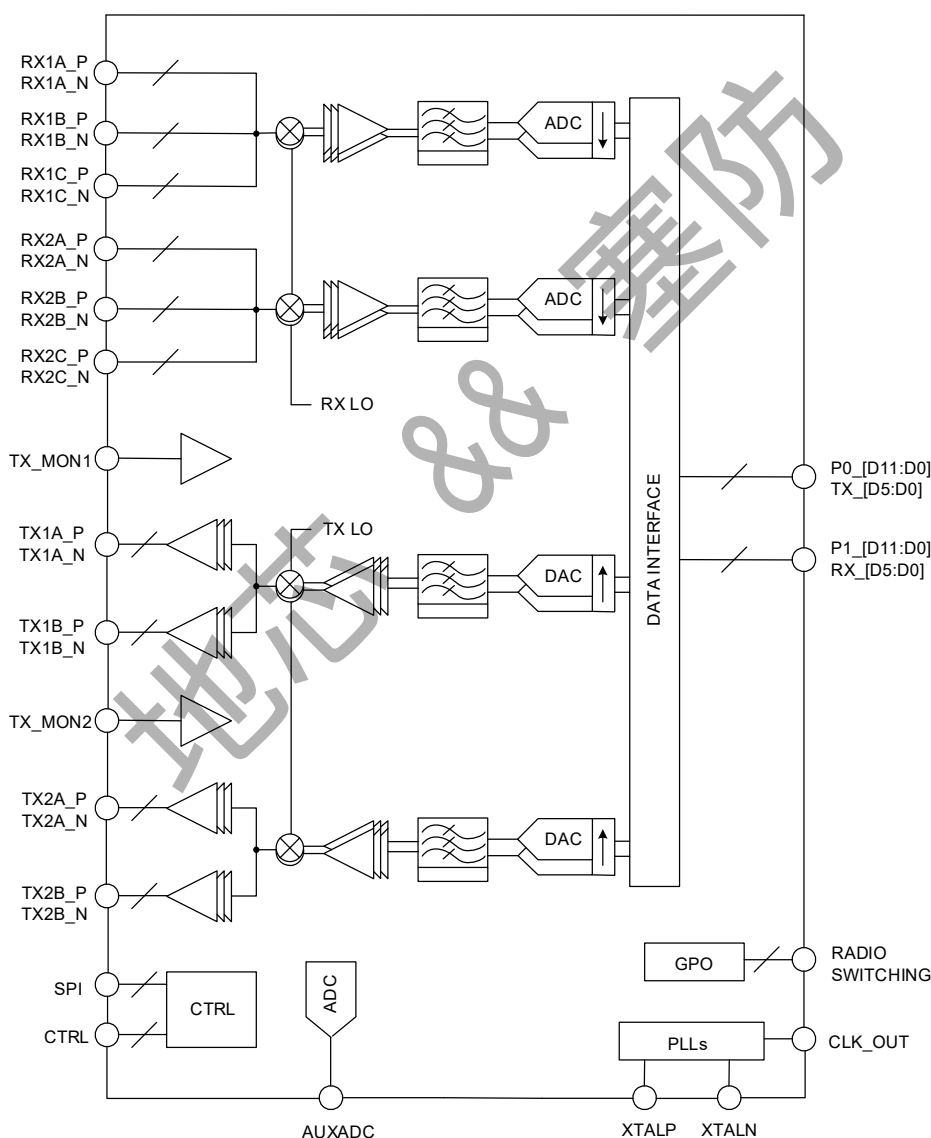


图 1-1

GC080X 的内部框图如图 1-2 所示

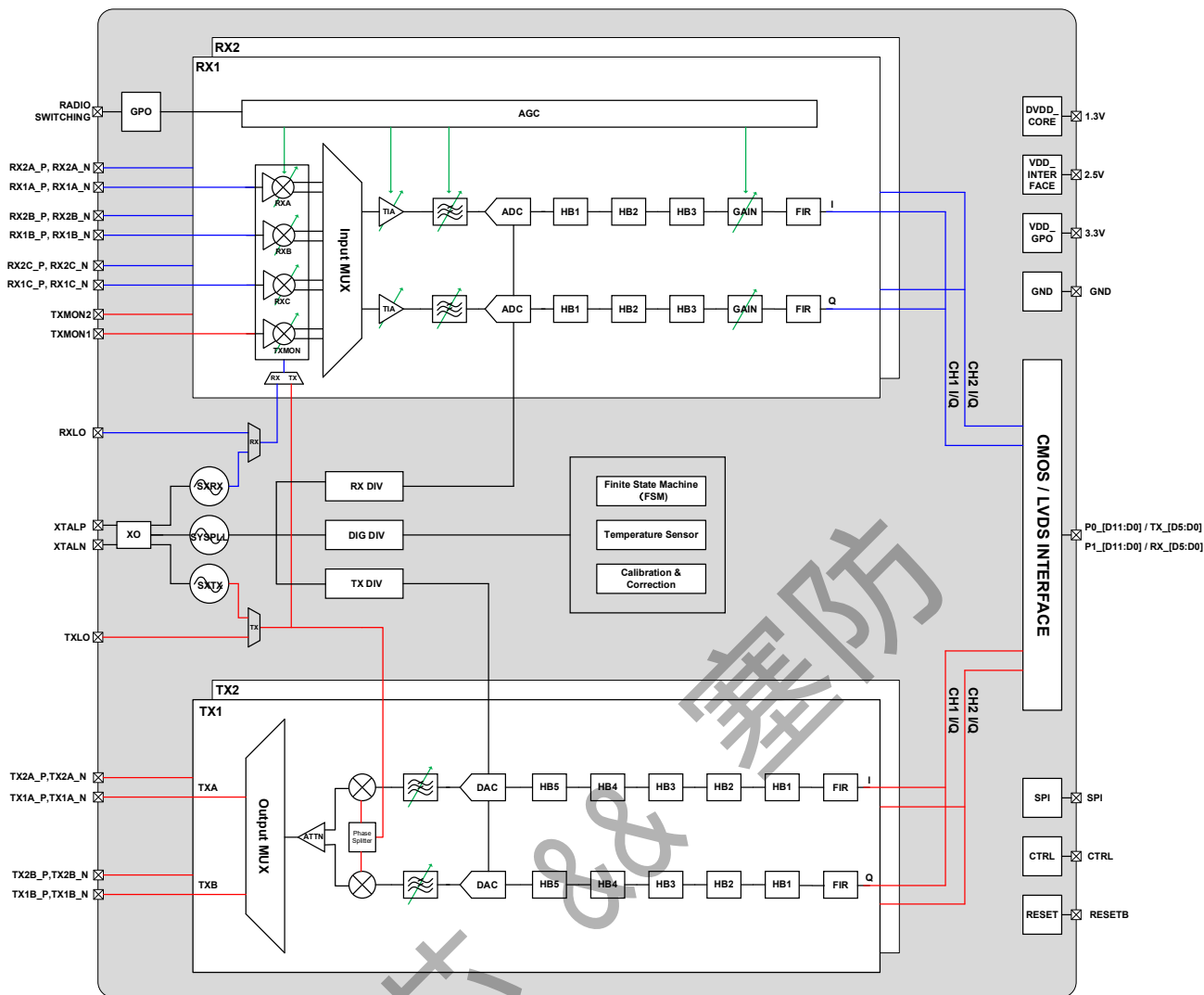


图 1-2

2、原理图设计要求：

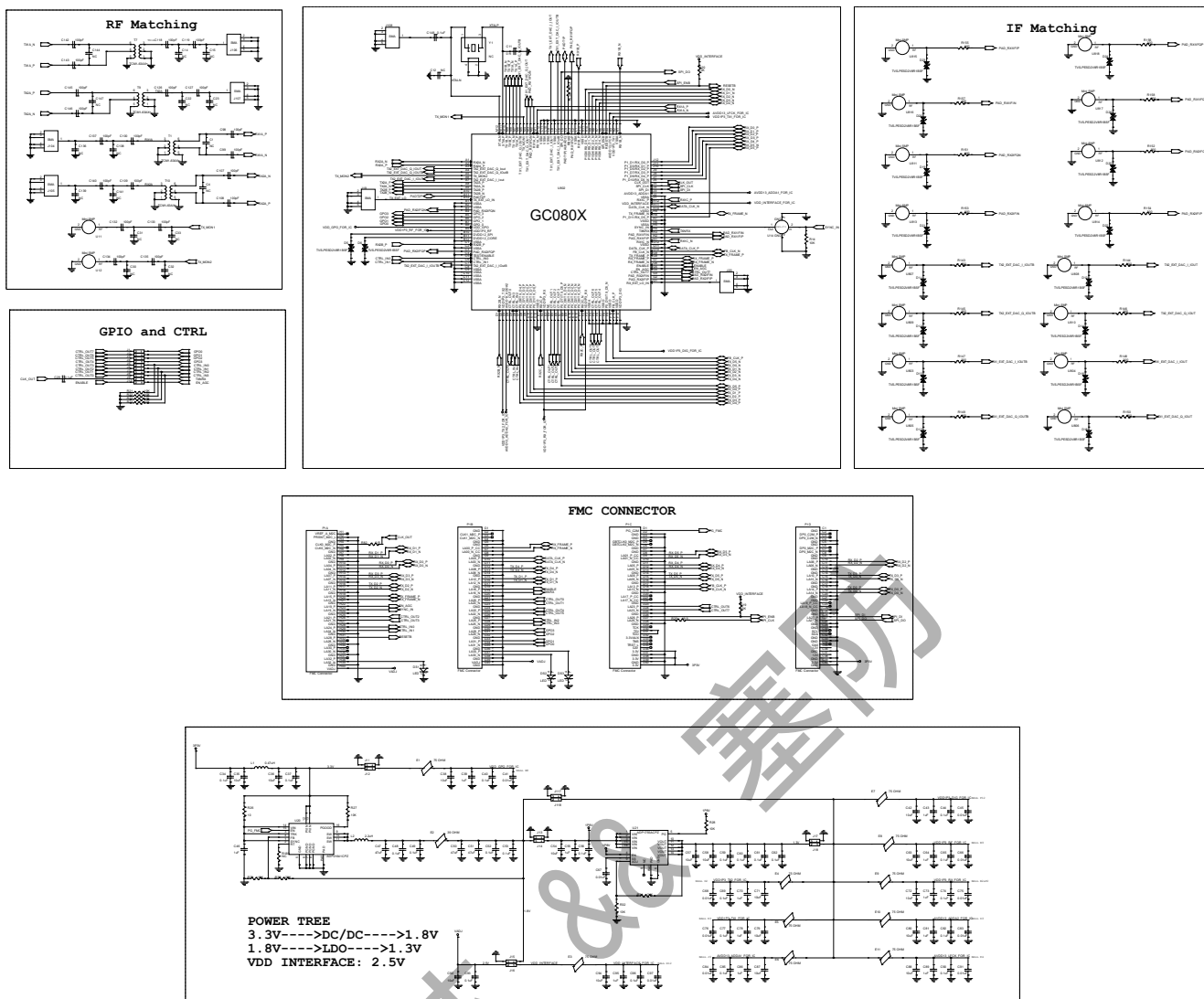


图 2-1

2.1、供电部分

数字供电部分

VDD_GPO 的供电电压为 3.3V

VDD1P3_DIG 的供电电压为 1.3V (实际可以接受的范围是 1.3V~1.5V 之间)

VDD_INTERFACE 的供电电压为 1.8V~2.5V 之间，推荐值用 2.5V

模拟供电部分

VDD1P3_RF、VDD1P3_TX1、VDD1P3_TX2、VDDD_13_AVDDA1、VDDD_13_AVDDA2、VDD1P3_RX、AVDD13_LFCK 的供电电压为 1.3V (实际可以接受的范围是 1.3V~1.5V 之间)

2.2、数字部分

SPI_ENB 需要接 10K 欧姆电阻上拉到 VDD_INTERFACE

SPI_CLK 串联 49.9 欧姆电阻到基带端

SPI_DI、SPI_DO 和基带端直连即可

P0、P1、TX_FRAME、DATA_CLK、FB_CLK、TXNRX、ENABLE、EN_AGC 和基带端直连即可

2.3、射频部分

RX 可以接单端 RX1A_P、RX1A_N、RX1B_P、RX1B_N、RX1C_P、RX1C_N、RX2A_P、RX2A_N、RX2B_P、RX2B_N、RX2C_P、RX2C_N 一共 12 个口，接单端需隔直电容；

RX 可以接差分 RX1A、RX1B、RX1C、RX2A、RX2B、RX2C 一共 6 对差分，接差分在巴伦和 GC080X 芯片之间需要接隔直电容；

TX 可以接差分 TX1A、TX1B、TX2A、TX2B、一共 4 对差分，接差分在巴伦和 GC080X 芯片之间需要接隔直电容；

TX_MON1、TXMON2 可直接外界射频信号，需隔直电容

TXEXT_LO_IN、RXEXT_LO_IN 为外灌本振信号，无需外部隔直电容

2.4、时钟部分

XTALN 可接单路的时钟信号，建议在输入端串联一个 0.1uF 电容。XTALP 建议串联一个 0.1uF 的电容到地。

采用无源晶体方案时，XTALP 引脚需要并联一个到地的 0 欧姆电阻短路，单独调整 XTALN 网络位置并联电容的大小即可。

2.5、其他部分

RBIAS 引脚需要串联一颗精度 1% 的 14.3K 欧姆电阻到地

RESETB 需要接 10K 欧姆电阻上拉到 VDD_INTERFACE

VSSA 为模拟地，VSSD 为数字地

3、软件配置：

GC080X 在 Linux/no-os 系统中提供完整的驱动，在 Linux 系统中驱动采用 IIO 架构进行编写。下面我们以 Linux 中的 IIO 驱动接口的方式介绍 GC080X 的配置。Linux 中 IIO 接口的命名规则是 dir_type[x]_function，dir 表示方向，有 in(RX)和 out(TX)，type 有 altvoltage 和 voltage 等，x 表示 channel，function 表示该接口的功能。

3.1、初始化和校准

概述

每次 GC080X 重新上电后，芯片复位后 SPI 总系已就绪，主控通过 SPI 总线对芯片进行配置。在 GC080X 能够正常工作之前，必须先让内部的 LDO 使能，可以往 fsm_test 接口写"power_up"来使能内部的 LDO，该接口对应的函数为 chip_power_up；初始化 RF 和 ADC/DAC 等模拟电路，往 fsm_test 接口写"analog_setup"，该接口对应的函数为 chip_set_up。为了使得 GC080X 工作在良好的状态，需要做初始化校准，往 calib_mode 接口中写入下列校准项。初始化校准项和对应参数说明请参见表 3.1.1。

校准项	参数
r	芯片模拟电路基本校准，无参数
txdac	tx 链路中的 DAC 校准，在使用 DAC 之前，需要做此校准，参数：0 为 channel1,1 为 channel2
tx_dc_offs	tx 直流偏移校准，参数：0 为 channel1,1 为 channel2
rx_bw	rx 的 bandwidth 校准，在 rx 开始接收数据之前可以进行此校准，参数：通道，0 为 channel1,1 为 channel2；带宽，可选 BT,LTE5,LTE10,LTE20,LTE40,LTE60,LTE80,LTE100；接收端口，可选 A_P,A_N,A_BALANCED,B_P,B_N,B_BALANCED,C_P,C_N,C_BALANCED
rx_dc_offs	rx 直流偏移校准，参数：通道，0 为 channel1,1 为 channel2；接收端口，可选 A_P,A_N,A_BALANCED,B_P,B_N,B_BALANCED,C_P,C_N,C_BALANCED；LO，RX 当前工作频率

表 3.1.1

3.2、参考时钟配置和 BBPLL

GC080X 包含一个独立的 SYSPLL，用于产生 ADC/DAC 的采样时钟，基带采样时钟以及 GC080X 内部使用的时钟。GC080X 能够支持的参考时钟频率范围为 10MHz 到 80MHz 之间。驱动默认配置外部参考时钟频率为 30.72MHz，如果需要修改别的频率，可往 syspll_freq 接口中写入期望值（单位是 Hz），该接口对应的函数为 syspll_band_selection。

3.3、RFPLL 配置

GC080X 包含两个独立的 RFPLL，SXTX 以及 SXRX，其中 SXTX 可用于 SXRX。配置 TX1 LO 往 out_altvoltage0_TX_LO_frequency 接口写入频率值，比如 2400000000Hz，该接口对应的函数 tx_band_selection。配置 RX1 LO 往 in_altvoltage0_RX_LO_frequency 接口写入频率值，比如 2300000000Hz，该接口对应的函数 rx_band_selection。当 TX 和 RX 的频率很接近时，SXTX 可以选用 Core2，用于解决相互之间的 VCO pulling，但 SXTX 的 Core2 能支持的射频输出频率范围为 800MHz 到 3.5GHz，当使用 SXTX 的 Core2 时，需要往 out_TX_LO_C2 接口写入频率值，比如 2400000000Hz，该接口对应的函数 tx1_band_selection_c2。

3.4、外部本振使用配置

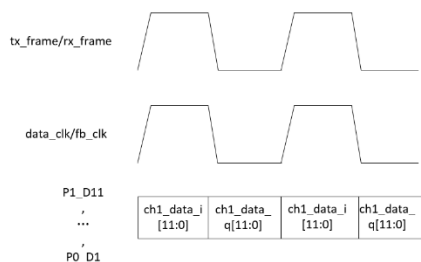
GC080X 的 TX LO 和 RX LO 可以使用外部本振输入，但此时要求外部输入信号频率是 TX LO 或者 RX LO 的两倍。当使用外部本振输入时，比如 TX1，往 out_altvoltage0_ext_TX_LO_frequency 写入频率值，比如 2400000000Hz。同样，外部本振方案情况下，TX LO 也可以用于 RX LO。

3.5、射频端口配置

GC080X 的 TX 有 4 对差分输出，分别对应 TX1A/TX1B/TX2A/TX2B 这 4 组端口。RX 有 6 对差分输入，分别对应 RX1A/RX1B/RX1C/RX2A/RX2B/RX2C 这 6 组端口。RX 也可以当做单端来使用，此时有 12 个端口，分别对应 RX1A_P/RX1A_N；RX1B_P/RX1B_N；RX1C_P/RX1C_N；RX2A_P/RX2A_N；RX2B_P/RX2B_N；RX2C_P/RX2C_N。可以读取 in_voltage_rf_port_select_available 查看 RX 的可用端口，比如配置 RX1 端口为 A_BALANCED，可以往 in_voltage0_rf_port_select 接口中写入 A_BALANCED，该接口对应的函数为 set_rx_port。可以读取 out_voltage_rf_port_select_available 查看 TX 的可用端口，比如配置 TX1 端口为 A，可以往 out_voltage0_rf_port_select 接口中写入 A，该接口对应的函数为 set_tx_port。

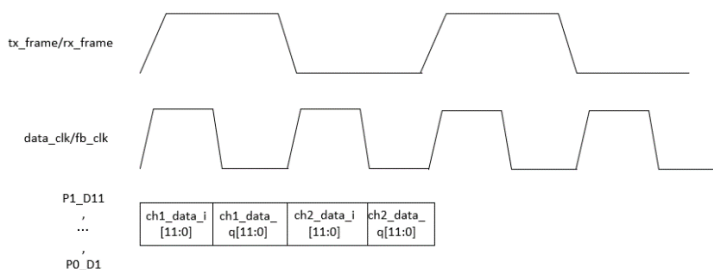
3.6、数字接口配置

GC080X 通过数字接口与外围器件（FPGA/ASIC）通过数据接口传输基带数据，时序图如下。TDD LVDS 1R1T 模式下接口时序如图：



TDDTX 采用 TX_FRAEM 同步，TDDRX 采用 RX_FRAME 同步。

TDD LVDS 2R2T 模式下接口时序如图：



TDDTX 采用 TX_FRAEM 同步，TDDRX 采用 RX_FRAME 同步。

FDDLVDs 模式如下：

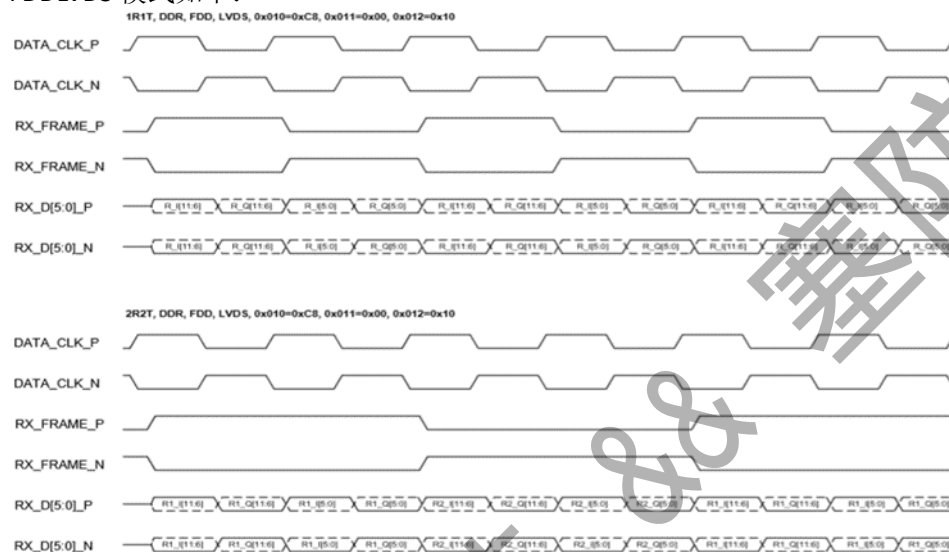


Figure 17: Receive Data Path, LVDS

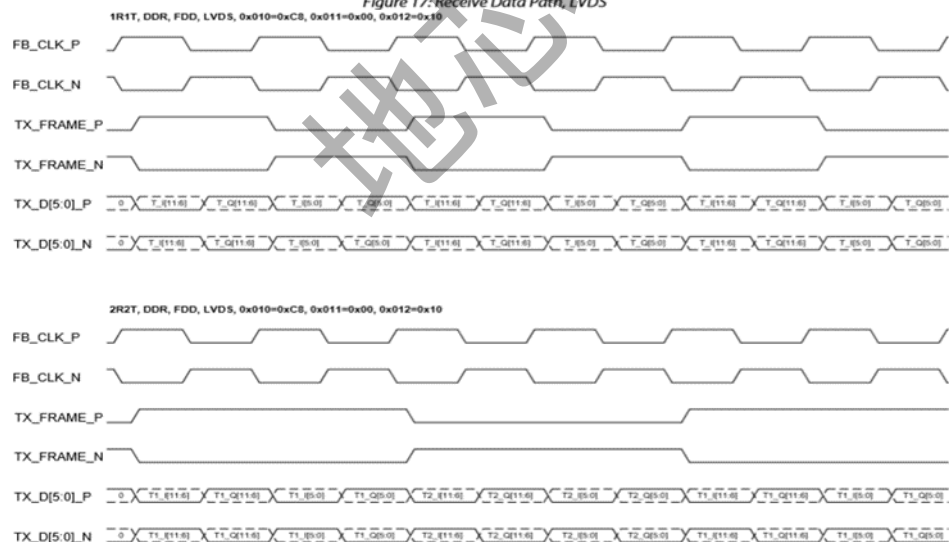


Figure 18: Transmit Data Path, LVDS

如有其它接口传输配置需求，可能不在上述波形图范围之内。可视客户具体需求给出。

接口寄存器介绍如下，具体使用请使用 API

	绝对地址	复位值	Bit[7:0]	信号描述
PORT_CTRL_0	061	00	7	TX_SWAP_IQ
			6	RX_SWAP_IQ
			5	TX_SWAP_CHANNEL

			4	RX_SWAP_CHANNEL
			3	RX_FRAME_PULSE_MODE
			2	RESERVED
			1	RESERVED
			0	RESERVED
PORT_CTRL_1	062	00	7	RESERVED
			6	RESERVED
			5	RESERVED
			4	INVERT_RX1
			3	INVERT_RX2
			2	INVERT_TX1
			1	INVERT_TX2
			0	RESERVED
PORT_CTRL_2	063	00	7	RESERVED
			6	SWAP_PORTS
			5	SINGLE_DATA_RATE
			4	LVDS_MODE
			3	HALF_DUPLEX_MODE
			2	SINGLE_PORT_MODE
			1	FULL_PORT
			0	RESERVED
CHN_CTRL_0	0DC	60	3	RX2_EN
			2	RX1_EN
			1	TX2_EN
			0	TX1_EN

表 3.6.1

以上均为 RW

061: bit7 1: TX 所有通道 IQ 数据位置互换
 bit6 1: RX 所有通道 IQ 数据位置互换
 bit5 1: TX1 和 TX2 数据交换
 bit4 1: RX1 和 RX2 数据交换
 bit3 1: 目前模式请都配置为 1

062: bit4~bit1 1: 各个通道传输数据*-1

063: bit6 1: P0 与 P1 端口互换, 包括传输信号方向, 按实际使用场景决定
 Bit5 1: SDR 模式 0: DDR 模式
 Bit4 1: LVDS 模式 0: CMOS 模式
 Bit3 1: TDD 模式 0: FDD 模式
 Bit2 1: single port 模式 0: dual port 模式
 Bit1 1: full port 模式

0DC bit3~bit0 1: 对应通道使能, 比如 FDD 1R1T, TX1enable RX1enable, 则为 05。比如 TDD 1R1T, TX2enable, TX1enable, 则为 0a。比如 TDD2R2T, TX12enable, RX12enable, 则为 0f。具体使用请使用 API

接口 DATA_CLK 的配置寄存器是 0x0DE, 此寄存器只能使用独热码配置, 切换时钟时, 可以直接切换独热码, 支持动态无毛刺切换频率。该寄存器描述如表 3.6.2 所示。

DATA_CLK_SEL_0	0x0DE	BIT7	RESERVED
		BIT6	DATA_CLK6_SEL

		BIT5	DATA_CLK5_SEL
		BIT4	DATA_CLK4_SEL
		BIT3	DATA_CLK3_SEL
		BIT2	DATA_CLK2_SEL
		BIT1	DATA_CLK1_SEL
		BIT0	DATA_CLK0_SEL

表 3.6.2

接口 DATA_CLK 时钟频率的配置与数字滤波器组，接口配置，制式紧密相关，具体使用请参考 API

3.7、数字滤波器配置

如下图 3.7.1 所示为 TRX 链路上的数字滤波器组。滤波器组，接口模式，数字系统时钟模块（数字的 LO），按不同通讯制式配置较为复杂，部分配置存在 lut 中，部分配置不在 lut 中，数字滤波器不建议单独配置，软件已配置。

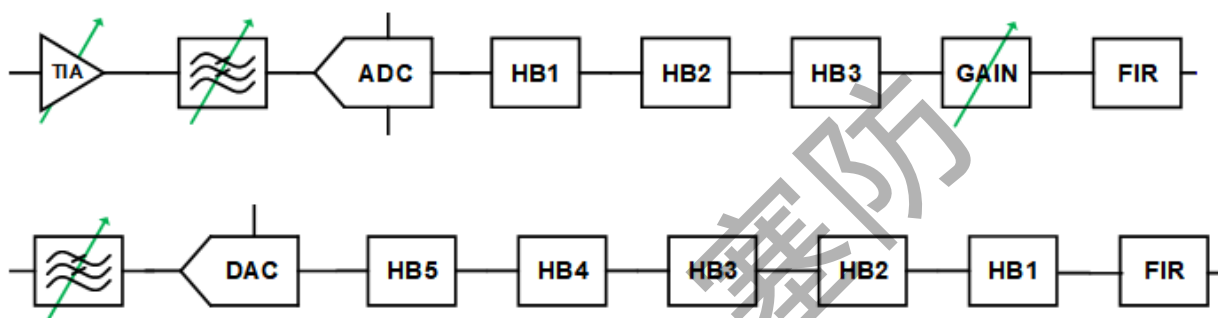


图 3.7.1

数字接受通路示意如上，介绍如下，具体使用请参考 API

	绝对地址	复位值	Bit[7:0]	信号描述
TX1_FLT_CTRL_0	001	00	7	RESERVED
			6	TX1_FIR_EN
			5	RESERVED
			4	TX1_CIC_EN
TX2_FLT_CTRL_0	002	00	7	RESERVED
			6	TX2_FIR_EN
			5	RESERVED
			4	TX2_CIC_EN
RX1_FLT_CTRL_0	003	00	7	RX1_HB3_EN
			6	RX1_HB2_EN
			5	RX1_HB1_EN
			4	RX1_FIR_EN
RX2_FLT_CTRL_0	006	00	7	RX2_HB3_EN
			6	RX2_HB2_EN
			5	RX2_HB1_EN
			4	RX2_FIR_EN

以上控制信号如信号描述所示，1 表示 enable，0 表示 disable，以上均为 RW

3.8、增益控制方法

GC080X 包含两条相互独立的 RX 通道，每条内部接收链路最大可提供 43dB 的模拟增益可调节范围（10dB~52dB），其中 30dB 位于模拟滤波器之前（称为 RFFE 增益），12dB 位于模拟滤波器之后（称为 BBF 增益）。若有需要也可开启数字增

益，数字增益上限为 50dB，精度为 0.25dB。

本章节将按照下面的步骤阐述 GC080X 的增益控制方法：

1、GC080X 的增益分配方式。GC080X 采用两种分配形式，即 full table 模式和 split table 模式。在不同增益控制方法下体现为需要增益调整时进行统一调整，还是区分 RFFE 和 BBF 增益分开调度。

2、信号峰值和平均功率检测器。GC080X 在每条接收链路中提供了一个位于在 BBF 前的峰值包络检测器 (LMT Peak Detector)，一个基于 ADC 输出的峰值检测器(ADC Peak Detector)，一个基于 HB1 输出的平均功率检测器。它们将为增益控制提供依据，相关配置需要在进入正常接收前设置好。

3、Ctrl-Out PIN 的设定。无论 GC080X 采用哪种增益控制方法，都可能需要从 Ctrl-Out 管脚的输出获得实时有用信息，尤其是手动控制的方法。GC080X 提供了多组 Ctrl-Out 的排列方式，在方案集成设计初期往往需要提前考虑。

4、GC080X 手动增益控制方法。GC080X 提供基于 SPI 或 CTRL-IN PIN 两种手动增益控制的方式。

5、GC080X 自动增益控制方法。GC080X 提供了一种自动增益控制的机制，具备遭遇大阻塞信号时快速衰减，在信号幅度平稳时慢慢收敛的行为特性。

3.8.1、GC080x 增益分配方式

GC080X 的 RX 通道中包含 RFFE 增益和 BBF 增益。其中 RFFE 增益位于模拟滤波器之前，最大 30dB，步进为 6dB。其中 BBF 增益位于模拟滤波器之后，最大 13dB，步进为 1dB。GC080X 包含两种对模拟增益进行分配的方式，一种为 full table 模式，即调节增益的时候不区分是 RFFE 增益还是 BBF 增益，而仅就事先保存的增益分配表对增益进行分配。对于增益的增大或减小，则体现为对这张表的读地址进行调整。另一种为 split table 模式，在进行增益控制时根据位于 RX 链路上不同检测器的输出信息具体决定调整 RFFE 增益还是 BBF 增益，且它们的步进通常可以根据需求单独设定。

如果用户倾向于简单有效的增益控制，可使用 full table 模式。若存在较强的带外干扰且对信号信噪比敏感的应用需求，则可以使用 split table 模式。

3.8.2、Full table 模式

采用 full table 模式需要提前将 RFFE 增益和 BBF 增益分配到一张总表中，需满足的规则是：每个增益挡位步进为 1dB，增益排列从小至大。因此在增益变化时，对于增益的增减操作将转换为对该表读地址的加减。

下表 3.8.2 是一种符合规则的增益表分布。

Ext LNA	RFFE 增益	BBF 增益	增益表地址	实际增益
0	0	0	0	12
0	0	1	1	13
0	0	2	2	14
0	0	3	3	15
0	0	4	4	16
0	0	5	5	17
0	1	0	6	18
0	1	1	7	19
0	1	2	8	20
0	1	3	9	21
0	1	4	10	22
0	1	5	11	23
0	2	0	12	24
0	2	1	13	25
0	2	2	14	26
0	2	3	15	27
0	2	4	16	28
0	2	5	17	29
0	3	0	18	30

0	3	1	19	31
0	3	2	20	32
0	3	3	21	33
0	3	4	22	34
0	3	5	23	35
0	4	0	24	36
0	4	1	25	37
0	4	2	26	38
0	4	3	27	39
0	4	4	28	40
0	4	5	29	41
0	5	0	30	42
0	5	1	31	43
0	5	2	32	44
0	5	3	33	45
0	5	4	34	46
0	5	5	35	47
0	5	6	36	48
0	5	7	37	49
0	5	8	38	50
0	5	9	39	51
0	5	10	40	52
0	5	11	41	53
0	5	12	42	54

上表中前三列是存储在芯片中的信息，每一个地址 8bit，bit7 存放外置 LNA 使能或者 bypass，bit6~bit4 存放 RFGE 增益，bit3~bit0 存放 BBF 增益。Full table 最大可以存储 128 个地址，但实际可能没有这么大的控制范围需求，因此需要设定当前增益表的深度。如果当前增益表模式选择为 full table 模式，则需要在上电后至少初始化一次该表中的内容，推荐在 wait 状态下对该表进行设置。在出厂检验时，会对每个增益档位进行校准测试，因此增益排布可能并非如上图所示顺序，最终用户也可根据实际需要定制化这张表。

此外还需要针对第一次进入 RX 后的初始 full table 增益进行设定。

具体配置方法参考 API 函数 gctrl_full_table_setting(ch, pstGainTableFull) 中的说明。

3.8.3、Split table 模式

Split table 模式下若要求增益变化，用户或芯片内部会根据一定的规则将需要调整的增益分配到 RFFE 增益和 BBF 增益上去。

对于 AGC 而言，则会按照下面的调整规则分配增益：

1、LMT 峰值检测器提示饱和，需要降低增益，则会降低 RFFE 增益而非降低 BBF 增益。

2、ADC 峰值检测器提示饱和，需要降低增益，则根据用户设置的优先级规则决定修改增益在哪个增益上。

3、平均功率检测器根据计算增益收敛目标和当前信号功率的差值，需要增加或者降低增益时，若 LMT 峰值检测器告知信号水平很大，不允许增加，则增加增益在 BBF 增益上。若 ADC 峰值检测器告知信号水平很大，不允许增加，则模拟增益将不会改变。若这两个检测器没有上述情况告知，根据用户设定的优先级规则决定修改在哪个增益上。

用户设置的优先级规则是指用户可指定在 RFFE 增益在增益变化后大于某个挡位时，优先增加 BBF 增益，降低 RFFE 增益，否则优先增加 RFFE 增益，降低 BBF 增益。比如说当前 RFFE 增益挡位为 3，BBF 增益挡位为 6，用户可设定 RFFE 增益大于 3 的时候优先增加 BBF 增益。现在要增加 6dB，若加在 RFFE 后挡位为 4，将触发优先级规则。因此最终收敛结果会是 RFFE 挡位为 3，BBF 增益挡位为 12。

对于 MGC 而言，可以由用户选择决定修改增益在哪部分增益上，也可以仍然按照芯片内部如 AGC 一样的调整规则自动调整增益对象。

此外还需要针对第一次进入 RX 后的初始 split table 增益进行设定。

具体配置方法见 API 函数 `gctrl_split_table_setting(ch,pstGainTableSplit)` 中的描述。

3.8.4、信号峰值和平均功率检测器

GC080X 在每条接收链路中提供了一个位于在 BBF 前的模拟的峰值包络检测器 (LMT Peak Detector)，一个基于 ADC 输出的峰值检测器(ADC Peak Detector)，一个基于 HB1 输出的平均功率检测器。位于 RX 链路三个位置的信号峰值或平均功率检测器将为增益控制提供依据，需要在开启增益控制前配置好。

3.8.4.1、LMT 峰值检测器

LMT 峰值检测器位于混频器之后、模拟滤波器之前。注意到 LMT 峰值检测器若开启将占用一个 AUX ADC 的硬件资源。最终 LMT 峰值检测器将判断出 2 种情况：

(1) 信号幅度过大，则后级 AGC 电路应考虑降低前级增益。此时 LMT 峰值检测器将送出标志位 `flg_lmt_overload` 到增益控制器和 Ctrl-Out 管脚。

(2) 信号幅度比较大，虽然没有达到需要降低前级增益的程度，应通知后级电路此时信号幅度已经比较大，无须继续增加增益。此时 LMT 峰值检测器将送出标志位 `flg_lmt_prevent_gain_inc` 到增益控制器和 Ctrl-Out 管脚。

这两种情况最终判别是否发生，除了瞬时信号的幅度值门限作为第一依据，还可就超过某个门限的次数作为第二判别依据。

LMT 峰值检测器的相关配置需要在进入 RX 前提前配置好，具体配置参数见 API 函数 `gctrl_lmt_pdt_setting(ch,pstLmtPdt)` 的说明。

3.8.4.2、ADC 峰值检测器

ADC 峰值检测器位于 ADC 后面，基于 ADC 的采样输出判断是否需要告警。将提供四种情况的判断，它们分别是：

（1）峰值过大，需要快速降低前级模拟增益。此时 ADC 峰值检测器将送出标志位 `flg_adc_overload_large` 到增益控制器和 `Ctrl-Out` 管脚。

（2）峰值仍然很大，需要降低前级模拟增益。此时 ADC 峰值检测器就送出标志位 `flg_adc_overload_small` 到增益控制器和 `Ctrl-Out` 管脚。

（3）峰值没有超出可承受范围，虽然没有达到需要降低前级增益的程度，应通知后级电路此时信号幅度已经比较大，无需继续增加增益。此时 ADC 峰值检测器就送出标志位 `flg_adc_prevent_inc` 到增益控制器和 `Ctrl-Out` 管脚

（4）峰值过小，需要提高前级增益。通知后级电路此时信号幅度过小，无须继续增加增益。此时 ADC 峰值检测器就送出标志位 `flg_adc_underload` 到增益控制器和 `Ctrl-Out` 管脚。

针对前三种情况，都设计为连续检测多个 ADC 数据超过门限指定数目之后将相关标志位 `flg_adc_overload_large`、`flg_adc_overload_small`、`flg_adc_prevent_inc` 置为 1。

对于第四种情况，则设计为检测多个 ADC 数据少于门限指定数目之后，将标志位 `flg_adc_underload` 置位为 1。

3.8.4.3、HB1 平均功率检测器

HB1 数字功率检测器位于最后一级 HBF 后面，FIR 前面。它将根据预设的时间长度对 HB1 数据进行平方和累加。每得到一个有效统计值后，将转换为 dBFs 值，并置 `flg_avp_valid` 标志位为 1，告知当前可提取一个有效功率统计值。注意该功率统计值并不包含数字增益的结果，因此即便开启数字增益，在寄存器或者 `Ctrl-Out` 管脚读取的平均功率是施加数字增益前的平均功率。

用户可根据需求设定参与功率统计的点数，具体配置方法见 API 函数 `gctrl_hbf_avp_setting(ch,ena, pstAdcPdt)`。

3.8.5、Ctrl-Out PIN

GC080X 包含了 12 根 PIN 管脚输出指定信息，可通过操作寄存器或 API 函数的方式调整输出的信息内容。与增益控制相关的管脚输出内容为：

Register	Control output bit position											
	CTRL-OUT D7	CTRL-OUT D6	CTRL-OUT D5	CTRL-OUT D4	CTRL-OUT D3	CTRL-OUT D2	CTRL-OUT D1	CTRL-OUT D0	GPO-OUT D3	GPO-OUT D2	GPO-OUT D1	GPO-OUT D0
18	RX1 lmt overload	RX1 adc overload lg	RX1 adc overload sm	RX1 hbf overflow	RX1 hbf low power	RX1 hbf avp valid	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 lmt underload	RX1 adc underload	RX1 ext lna ena	RX2 ext lna ena
19	RX2 lmt overload	RX2 adc overload lg	RX2 adc overload sm	RX2 hbf overflow	RX2 hbf low power	RX2 hbf avp valid	RX2 lmt prevent inc	RX2 adc prevent inc	RX2 lmt underload	RX2 adc underload	RX1 ext lna ena	RX2 ext lna ena
20	RX1 lmt overload	RX1 adc overload lg	RX1 hbf overflow	RX1 hbf low power	RX2 lmt overload	RX2 adc overload lg	RX2 hbf overflow	RX2 hbf low power	RX1 hbf avp valid	RX2 hbf avp vld	RX1 ext lna ena	RX2 ext lna ena
21	RX1 lpf index[3]	RX1 lpf index[2]	RX1 lpf index[1]	RX1 lpf index[0]	RX1 lmt index[2]	RX1 lmt index[1]	RX1 lmt index[0]	RX1 gain change	RX1 rssi symbol vld	RX1 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
22	RX2 lpf index[3]	RX2 lpf index[2]	RX2 lpf index[1]	RX2 lpf index[0]	RX2 lmt index[2]	RX2 lmt index[1]	RX2 lmt index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
23	RX1 full index[6]	RX1 full index[5]	RX1 full index[4]	RX1 full index[3]	RX1 full index[2]	RX1 full index[1]	RX1 full index[0]	RX1 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
24	RX2 full index[6]	RX2 full index[5]	RX2 full index[4]	RX2 full index[3]	RX2 full index[2]	RX2 full index[1]	RX2 full index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
28	RX1 avp[6]	RX1 avp[5]	RX1 avp[4]	RX1 avp[3]	RX1 avp[2]	RX1 avp[1]	RX1 avp[0]	RX1 avp vld	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena
29	RX2 avp[6]	RX2 avp[5]	RX2 avp[4]	RX2 avp[3]	RX2 avp[2]	RX2 avp[1]	RX2 avp[0]	RX2 avp vld	RX2 lmt prevent inc	RX2 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena

用户可在进入 RX 前设定好 CTRL-OUT 管脚的内容，也可以实时修改，但注意修改后需要延迟少许时间后端口信号才能稳定。

3.8.6、MGC 模式

MGC 有两种基本的控制方式，即通过 SPI 模式和通过外部管脚调整增益。这些具体设定的参数列表，请参考 API 函数 `gctrl_mgc_setting(ch,pstMgc)` 中的描述。

3.8.6.1、SPI 模式

当前若是 full table 模式，则直接设定 full table 的地址即可，注意 full table 的内容须要在进入接收前设定好。

若当前是 split table 模式，则可以直接对外置 LNA（若存在），RFFE 增益、BBF 增益进行单独设置。

具体设置方法请参考 API 函数 `gctrl_set_split_gain(ch,pstSplitGain)` 和 `gctrl_set_full_gain(ch,pstFullGain)` 的说明。

3.8.6.2、PIN 模式

PIN 模式下用户可通过线控快速对增益进行控制，GC080X 提供了 4 根控制输入用于 PIN 模式下的增益控制，分别是 CTRL-IN0, CTRL-IN1, CTRL-IN2, CTRL-IN3。它们的作用如下表所示：

管脚名称	作用
CTRL-IN0	增大 RX1 增益
CTRL-IN1	减小 RX1 增益
CTRL-IN2	增大 RX2 增益
CTRL-IN3	减小 RX2 增益

若当前采用 full table，上述调整的增益将直接作用于 full table 的读地址，从而改变增益。

若当前采用 split table，由于可调整的增益包括 RFFE 增益和 BBF 增益，而每个 RX 通道只有两根信号线可控制增益增减，因此需要区分调整何种增益变化。客户可选择通过 SPI 设定对何种增益进行控制，也可以交由 GC080X 判断哪部分增益进行控制。若交由 GC080X 进行判断，判断依据如小节《3.8.3 split table 模式》中所述规则所述。两种方式的调整步长都可以通过 API 函数 `gctrl_mgc_setting(ch,pstMgc)` 进行设定。

3.8.7、Slow Attack AGC 模式

GC080X 提供了一种对模拟增益的自动控制的方法，这里称为 Slow Attack AGC，适用于平稳连续信号，具有在面对突发大阻塞信号时快速降低前级增益，信号平稳时慢速收敛的行为特征。下面就 Slow attack agc 的行为进行描述，具体的参数列表请参考 API 函数 `gctrl_agc_slow_setting(ch,pstAgcSlow)` 中的描述。

3.8.7.1、快衰行为

在面临突发大阻塞信号或者信号强度突然变大的时候，信号峰值检测器若检测到饱和，会发出标志位信息告警。

若当前采用 full table，AGC 将马上降低 full table 的读地址，从而达到降低增益、避免饱和的目的。信号饱和可能导致 LMT 峰值检测器告知 `flg_lmt_overload`，ADC 峰值检测器告知 `flg_adc_overload_large`、`flg_adc_overload_small`，用户可选择采用不同的步进进行调整。

若当前采用 split table，AGC 将根据告警的峰值检测器自行判断应该优先降低哪一级的增益。若 LMT 峰值检测器告警，则会优先降低 RFFE 增益。若只有 ADC 峰值检测器告警，则会根据用户设置的优先级降低增益。这两种情况的增益调整步进都可以单独设定。

3.8.7.2、慢恢复行为

没有遭遇突发信号时，信号幅度相对平稳，此时也没有告警信息，AGC 将根据 HB1 平均功率检测器周期性的平均功率统计值和用户设定的收敛目标进行比较判断是否需要调整增益。

用户可以设置 4 条门限，这里被称为 `outter_high_th`，`outter_low_th`，`inner_high_th`，`inner_low_th`，单位是 dBfs。它们须满足 `outter_high_th > inner_high_th > inner_low_th > outter_low_th` 的关系。当平均功率检测器检测到信号功率位于 `inner_high_th` 和 `inner_low_th` 之间时将不会调整增益。当平均功率检测器检测到信号功率大于 `inner_high_th` 时将采用小步进调整值降低增益，当信号功率大于 `outter_high_th` 时将采用大步进调整值降低增益。当平均功率检测到信号功率小于 `outter_low_th` 时将采用大步进调整值降低增益，当信号功率小于 `inner_low_th` 时将采用大步进调整值增加增益。

3.8.7.3、增益更新时间

AGC 调整增益的时候用户可选收到峰值检测器告警或者平均功率统计结果后快速调整还是在指定周期计数时间到达的时候调整增益。所有快衰和慢恢复的行为都可以单独设置是否快速调整还是定时调整。

若选择快速调整，则 AGC 得到检测器信息后将尽可能快速计算出需要调整的增益量并调整。

若选择定时调整，则 AGC 可以开启一个增益周期计数器，周期由用户设定。当这个增益周期计数器结束的时候，AGC 同意按照上一个周期计数器最后得到的增益调整需求进行更新。

由于芯片外部和芯片内部的时钟可能是不同步的，用户可以通过对 `CTRL_IN1` 和 `CTRL_IN3` 拉起一个高脉冲的方式对增益周期计数器清零，从而达到同步的目的。

3.9、TX 功率控制配置

TX 功率控制的模拟精度是 1dB，默认配置如表 3.9 所示。只需要设定对应的 ATT 值即可；如果需要进行数字补偿，可进行 0.125dB/Step 精度的补偿。

ATT Index	Attenuation (dB)	PA Gain Index	BB Gain Index
0	0	0	0
1	1	0	1
2	2	0	2
3	3	0	3
4	4	0	4
5	5	0	5
6	6	1	3
7	7	1	4
8	8	1	5
9	9	2	3
10	10	2	4
11	11	2	5
12	12	3	3
13	13	3	4
14	14	3	5
15	15	4	3
16	16	4	4
17	17	4	5
18	18	5	3
19	19	5	4
20	20	5	5
21	21	6	3
22	22	6	4
23	23	6	5
24	24	7	3
25	25	7	4
26	26	7	5
27	27	8	3
28	28	8	4
29	29	8	5
30	30	8	6
31	31	8	7
32	32	8	8
33	33	8	9
34	34	8	10
35	35	8	11
36	36	8	12
37	37	8	13
38	38	8	14
39	39	8	15
40	40	8	16

表 3.9

3.10、CTRL OUT 相关输出

以下是 CTRL OUT 中与增益控制相关的位信息，使用者可根据需求将 CTRL OUT 选项调制合适的结果。

Register	Control output bit position											
	CTRL-OUT D7	CTRL-OUT D6	CTRL-OUT D5	CTRL-OUT D4	CTRL-OUT D3	CTRL-OUT D2	CTRL-OUT D1	CTRL-OUT D0	GPO-OUT D3	GPO-OUT D2	GPO-OUT D1	GPO-OUT D0
18	RX1 lmt overload	RX1 adc overload lg	RX1 adc overload sm	RX1 hbf overflow	RX1 hbf low power	RX1 hbf avp valid	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 lmt underload	RX1 adc underload	RX1 ext lna ena	RX2 ext lna ena
19	RX2 lmt overload	RX2 adc overload lg	RX2 adc overload sm	RX2 hbf overflow	RX2 hbf low power	RX2 hbf avp valid	RX2 lmt prevent inc	RX2 adc prevent inc	RX2 lmt underload	RX2 adc underload	RX1 ext lna ena	RX2 ext lna ena
20	RX1 lmt overload	RX1 adc overload lg	RX1 hbf overflow	RX1 hbf low power	RX2 lmt overload	RX2 adc overload lg	RX2 hbf overflow	RX2 hbf low power	RX1 hbf avp valid	RX2 hbf avp vld	RX1 ext lna ena	RX2 ext lna ena
21	RX1 lpf index[3]	RX1 lpf index[2]	RX1 lpf index[1]	RX1 lpf index[0]	RX1 lmt index[2]	RX1 lmt index[1]	RX1 lmt index[0]	RX1 gain change	RX1 rssi symbol vld	RX1 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
22	RX2 lpf index[3]	RX2 lpf index[2]	RX2 lpf index[1]	RX2 lpf index[0]	RX2 lmt index[2]	RX2 lmt index[1]	RX2 lmt index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
23	RX1 full index[6]	RX1 full index[5]	RX1 full index[4]	RX1 full index[3]	RX1 full index[2]	RX1 full index[1]	RX1 full index[0]	RX1 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
24	RX2 full index[6]	RX2 full index[5]	RX2 full index[4]	RX2 full index[3]	RX2 full index[2]	RX2 full index[1]	RX2 full index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
28	RX1 avp[6]	RX1 avp[5]	RX1 avp[4]	RX1 avp[3]	RX1 avp[2]	RX1 avp[1]	RX1 avp[0]	RX1 avp vld	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena
29	RX2 avp[6]	RX2 avp[5]	RX2 avp[4]	RX2 avp[3]	RX2 avp[2]	RX2 avp[1]	RX2 avp[0]	RX2 avp vld	RX2 lmt prevent inc	RX2 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena

表 3.10

3.11、控制输出

GPO 输出有哪些组合，如下图 3.11 所示为 CTRL-OUT 中与增益控制相关的位信息，使用者可根据需求将 CTRL OUT 选项调制合适的结果。

Register	Control output bit position											
	CTRL-OUT D7	CTRL-OUT D6	CTRL-OUT D5	CTRL-OUT D4	CTRL-OUT D3	CTRL-OUT D2	CTRL-OUT D1	CTRL-OUT D0	GPO-OUT D3	GPO-OUT D2	GPO-OUT D1	GPO-OUT D0
18	RX1 lmt overload	RX1 adc overload lg	RX1 adc overload sm	RX1 hbf overflow	RX1 hbf low power	RX1 hbf avp valid	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 lmt underload	RX1 adc underload	RX1 ext lna ena	RX2 ext lna ena
19	RX2 lmt overload	RX2 adc overload lg	RX2 adc overload sm	RX2 hbf overflow	RX2 hbf low power	RX2 hbf avp valid	RX2 lmt prevent inc	RX2 adc prevent inc	RX2 lmt underload	RX2 adc underload	RX1 ext lna ena	RX2 ext lna ena
20	RX1 lmt overload	RX1 adc overload lg	RX1 hbf overflow	RX1 hbf low power	RX2 lmt overload	RX2 adc overload lg	RX2 hbf overflow	RX2 hbf low power	RX1 hbf avp valid	RX2 hbf avp vld	RX1 ext lna ena	RX2 ext lna ena
21	RX1 lpf index[3]	RX1 lpf index[2]	RX1 lpf index[1]	RX1 lpf index[0]	RX1 lmt index[2]	RX1 lmt index[1]	RX1 lmt index[0]	RX1 gain change	RX1 rssi symbol vld	RX1 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
22	RX2 lpf index[3]	RX2 lpf index[2]	RX2 lpf index[1]	RX2 lpf index[0]	RX2 lmt index[2]	RX2 lmt index[1]	RX2 lmt index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
23	RX1 full index[6]	RX1 full index[5]	RX1 full index[4]	RX1 full index[3]	RX1 full index[2]	RX1 full index[1]	RX1 full index[0]	RX1 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
24	RX2 full index[6]	RX2 full index[5]	RX2 full index[4]	RX2 full index[3]	RX2 full index[2]	RX2 full index[1]	RX2 full index[0]	RX2 gain change	RX2 rssi symbol vld	RX2 rssi preamble vld	RX1 ext lna ena	RX2 ext lna ena
28	RX1 avp[6]	RX1 avp[5]	RX1 avp[4]	RX1 avp[3]	RX1 avp[2]	RX1 avp[1]	RX1 avp[0]	RX1 avp vld	RX1 lmt prevent inc	RX1 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena
29	RX2 avp[6]	RX2 avp[5]	RX2 avp[4]	RX2 avp[3]	RX2 avp[2]	RX2 avp[1]	RX2 avp[0]	RX2 avp vld	RX2 lmt prevent inc	RX2 adc prevent inc	RX1 ext lna ena	RX2 ext lna ena

图 3.11

3.12、状态机

GC080X 包含一个状态机，可实时控制设备的当前状态。状态机有两种控制方法，一个是 SPI 总线配置寄存器方式，另一个是 PIN 控制状态。如下图 3.12 所示为 TDD 和 FDD 在状态机中的状态。在芯片上电后，状态机处于 wait 状态，在状态机工作之前，我们首先需要设置 FSM_CTRL_0 寄存器 0x0CC，BIT7 设置为 1 表示通过 PIN 控制状态机，为 0 状态机通过 SPI 控制。

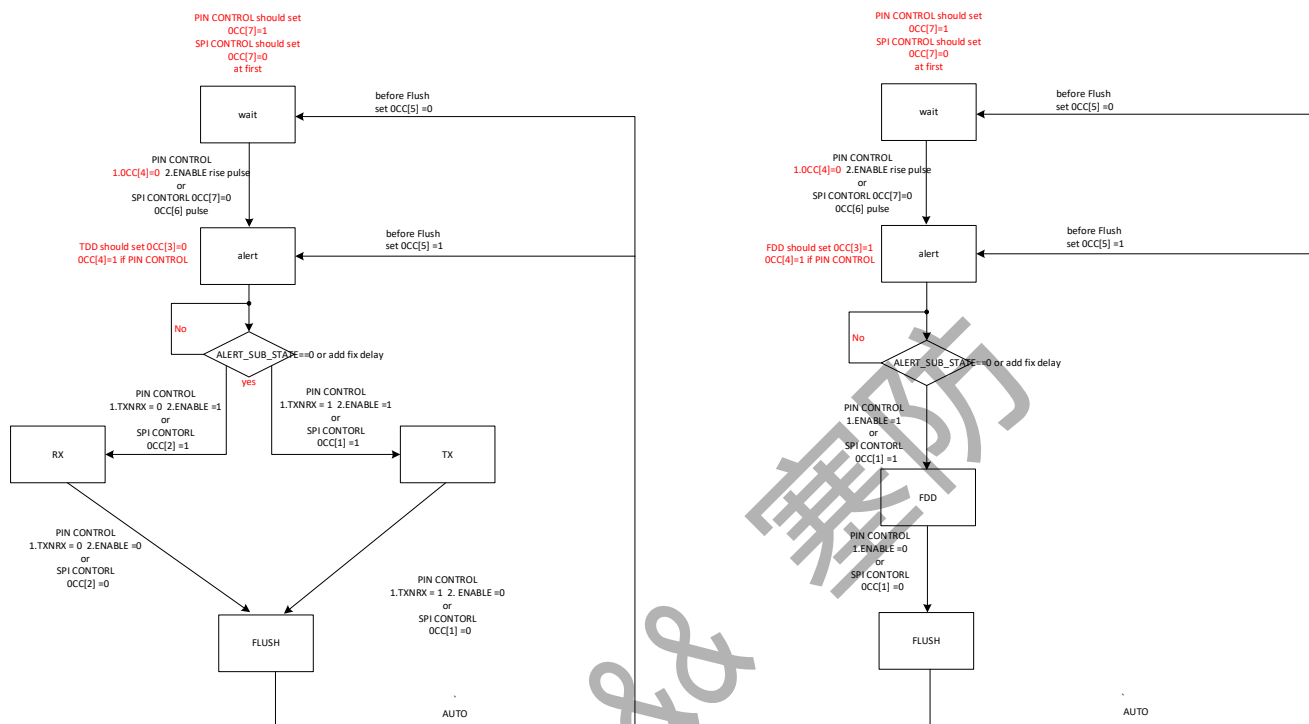


图 3.12

FSM_CTRL_0 寄存器是状态机基本控制寄存器，该寄存器的描述表如下表 3.12。

FSM_CTRL_0	0x0CC	BIT7	ENSM PIN_CTRL
		BIT6	FORCE_ALERT
		BIT5	TO_ALERT
		BIT4	LEVEL_MODE
		BIT3	FDD_MODE
		BIT2	FORCE_RX
		BIT1	FORCE_TX
		BIT0	LDO_FC_LOAD

表 3.12

在 TDD 模式下，可选的状态有 wait, alert, rx, tx, flush。在 wait 状态时，设置 FSM_CTRL_0 BIT6 一个脉冲信号（PIN 控制方式，BIT4 置为 0，ENABLE 上升沿脉冲），状态机进入 alert 状态。在 alert 状态时，TDD 模式下，FSM_CTRL_0 BIT3 为 0（PIN 控制方式，BIT4 置为 1）。当 alert 的子状态为 0 或只是简单延时，可控制状态机进入 RX 或 TX 状态，FSM_CTRL_0 BIT2 置 1（PIN 控制方式，TXNRX=0，ENABLE=1）进入 RX 状态；FSM_CTRL_0 BIT1 置 1（PIN 控制方式，TXNRX=1，ENABLE=1）进入 TX 状态。在 RX 状态下，FSM_CTRL_0 BIT2 置 0（PIN 控制方式，TXNRX=0，ENABLE=0）进入 flush 状态（相当于一个复位状态）。在 TX 状态下，FSM_CTRL_0 BIT1 置 0（PIN 控制方式，TXNRX=1，ENABLE=0）进入 flush 状态。在 flush 状态下，芯片执行完该状态下的任务，自动迁移到 alert（FSM_CTRL_0 BIT5 为 1）或 wait 状态（FSM_CTRL_0 BIT5 为 0）。

在 FDD 模式下，可选的状态有 wait, alert, FDD, flush。在 wait 状态时，设置 FSM_CTRL_0 BIT6 一个脉冲信号（PIN 控制方式，BIT4 置为 0，ENABLE 上升沿脉冲），状态机进入 alert 状态。在 alert 状态时，FDD 模式下，FSM_CTRL_0 BIT3 为 1（PIN 控制方式，BIT4 置为 1）。当 alert 的子状态为 0 或只是简单延时，可控制状态机进入 FDD 状态，FSM_CTRL_0 BIT1 置 1（PIN 控制方式，ENABLE=1）进入 FDD 状态。在 FDD 状态下，FSM_CTRL_0 BIT1 置 0（PIN 控制方式，ENABLE=0）进入 flush 状态。在 flush 状态下，芯片执行完该状态下的任务，自动迁移到 alert（FSM_CTRL_0 BIT5 为 1）或 wait 状态（FSM_CTRL_0 BIT5 为 0）。

(FSM_CTRL_0 BIT5 为 0)。

可以读取 ENSM_STATE_0 寄存器 0x0DA 来得知状态机所处状态。WAIT 000(default s/por state), ALERT 001, RX 011, TX 010, FDD 110, FLUSH 111。

3.13、软复位及 QEC 配置

数字接口软复位介绍如下：

	绝对地址	复位值	Bit[7:0]	信号描述
RESET_ENABLE_1	0F5	0f	7	RESERVED
			6	RESERVED
			5	RESERVED
			4	RESERVED
			3	RESERVED
			2	RESERVED
			1	BB_INTF_RSTN_EN
			0	RESERVED

0F5：软复位所在，bit1 为数字接口软复位，1 工作，0 复位，默认为 1 勿随意修改

QEC 部分接口如下，具体使用请参考 API

	绝对地址	复位值	Bit[7:0]	信号描述
CTRL_STT_0	200	00	7	RESERVED
			6	RESERVED
			5	tx_fiiq_comp_en
			4	tx_fdiq_comp_en
			3	RESERVED
			2	rx_fiiq_comp_en
			1	rx_fdiq_comp_en
			0	RESERVED

Bit5 1: TX1 FIIQ QEC Enale

Bit4 1: TX1 FDIQ QEC Enale

Bit2 1: RX1 FIIQ QEC Enale

Bit1 1: RX1 FDIQ QEC Enale

	绝对地址	复位值	Bit[7:0]	信号描述
CTRL_STT_1	A00	00	7	RESERVED
			6	RESERVED
			5	tx_fiiq_comp_en
			4	tx_fdiq_comp_en
			3	tx_keep
			2	rx_fiiq_comp_en
			1	rx_fdiq_comp_en

			0	goertzel_start
--	--	--	---	----------------

Bit5 1: TX2 FIIQ QEC Enale
Bit4 1: TX2 FDIQ QEC Enale
Bit2 1: RX2 FIIQ QEC Enale
Bit1 1: RX2 FDIQ QEC Enale

地芯科技 & 塞防

4、修订记录

版本	修改内容	修改时间	修改作者
GC080X 应用手册 V0.1	初版	2022.04.08	
GC080X 应用手册 V0.4	根据评审建议修改更新	2023.02.20	

地芯科技 & 塞防