



Gowin APB UART IP 用户指南

IPUG1209-1.0, 2025-04-30

版权所有 © 2025 广东高云半导体科技股份有限公司

GOWIN高云、**GOWIN**、、**GOWINSEMI**、**GOWIN**、Gowin、**高云**、晨熙、小蜜蜂、LittleBee、-V、GowinPnR、GoBridge 均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本文档内容的部分或全部，并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止反言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改文档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明
2025/04/30	1.0	初始版本。

目录

图目录.....	iii
表目录.....	iv
1 关于本手册.....	1
1.1 手册内容.....	1
1.2 相关文档.....	1
1.3 术语、缩略语.....	2
1.4 技术支持与反馈.....	2
2 概述.....	3
2.1	
2.2 功能特性.....	3
2.3 资源使用.....	3
3 功能描述.....	5
3.1 结构框图.....	5
3.2 发送器.....	5
3.3 接收器.....	6
3.4 波特率生成器.....	6
3.5 调制解调器控制器.....	7
3.6 回环模式.....	7
3.7 DMA 操作.....	8
4 信号描述.....	9
5 界面配置.....	11
6 编程模型.....	13
6.1 寄存器.....	13
6.1.1 寄存器概述.....	13
6.1.2 ID 和修订寄存器（0x00）.....	14
6.1.3 硬件配置寄存器（0x10）.....	14
6.1.4 过采样控制寄存器（0x14）.....	14
6.1.5 接收缓冲寄存器（当 DLAB = 0）（0x20）.....	15
6.1.6 发送保持寄存器（当 DLAB = 0）（0x20）.....	15
6.1.7 中断使能寄存器（当 DLAB = 0）（0x24）.....	15

6.1.8 分频锁存器最低有效字节（当 DLAB = 1）（0x20）	16
6.1.9 分频锁存器最高有效字节（当 DLAB = 1）（0x24）	16
6.1.10 中断识别寄存器（0x28）	16
6.1.11 FIFO 控制寄存器（0x28）	18
6.1.12 线路控制寄存器（0x2C）	19
6.1.13 调制解调器控制寄存器（0x30）	20
6.1.14 线路状态寄存器（0x34）	21
6.1.15 调制解调器状态寄存器（0x38）	22
6.1.16 暂存寄存器（0x3C）	23
6.2 驱动函数.....	23
6.2.1 驱动函数概述.....	24
6.2.2 apb_uart_get_capabilities.....	24
6.2.3 apb_uart_initialize.....	24
6.2.4 apb_uart_uninitialize.....	25
6.2.5 apb_uart_power_control.....	25
6.2.6 apb_uart_send.....	25
6.2.7 apb_uart_receive.....	26
6.2.8 apb_uart_transfer.....	26
6.2.9 apb_uart_get_txcount.....	26
6.2.10 apb_uart_get_rxcount.....	27
6.2.11 apb_uart_control.....	27
6.2.12 apb_uart_get_status.....	29
6.2.13 apb_uart_set_modem_control.....	29
6.2.14 apb_get_modem_status.....	30
6.2.15 apb_uart_irq_handler.....	30
6.2.16 apb_uart_dma_tx_event.....	30
6.2.17 apb_uart_dma_rx_event.....	31
7 编程序列.....	32
7.1 设置.....	32
7.2 发送数据.....	32
7.2.1 数据传输（不使用 DMA）	32
7.2.2 数据传输（使用 DMA）	33
7.3 接收数据.....	33
7.3.1 数据接收（不使用 DMA）	33
7.3.2 数据接收（使用 DMA）	33
8 参考设计.....	35

图目录

图 3-1 结构框图.....	5
图 3-2 回环模式中 MCR 和 MSR 的关系.....	8
图 5-1 界面配置.....	12

表目录

表 1-1 术语、缩略语.....	2
表 2-1 资源使用情况.....	3
表 3-1 时钟变化容差因子.....	7
表 4-1 信号描述.....	9
表 6-1 寄存器定义.....	13
表 6-2 ID 和修订寄存器.....	14
表 6-3 硬件配置寄存器.....	14
表 6-4 过采样控制寄存器.....	15
表 6-5 接收缓冲寄存器（当 DLAB = 0）.....	15
表 6-6 发送保持寄存器（当 DLAB = 0）.....	15
表 6-7 中断使能寄存器（当 DLAB = 0）.....	15
表 6-8 分频锁存器最低有效字节（当 DLAB = 1）.....	16
表 6-9 分频锁存器最高有效字节（当 DLAB = 1）.....	16
表 6-10 中断识别寄存器.....	17
表 6-11 中断控制类型.....	17
表 6-12 FIFO 控制寄存器.....	18
表 6-13 接收 FIFO 触发级别.....	19
表 6-14 发送 FIFO 触发级别.....	19
表 6-15 线路控制寄存器.....	19
表 6-16 奇偶校验位选择.....	20
表 6-17 调制解调器控制寄存器.....	20
表 6-18 线路状态寄存器.....	21
表 6-19 调制解调器状态寄存器.....	22
表 6-20 暂存寄存器.....	23
表 6-21 驱动函数定义.....	24
表 6-22 apb_uart_get_capabilities 函数定义.....	24
表 6-23 apb_uart_initialize 函数定义.....	25
表 6-24 apb_uart_uninitialize 函数定义.....	25
表 6-25 apb_uart_power_control 函数定义.....	25
表 6-26 apb_uart_send 函数定义.....	25

表 6-27 apb_uart_receive 函数定义.....	26
表 6-28 apb_uart_transfer 函数定义.....	26
表 6-29 apb_uart_get_txcount 函数定义.....	26
表 6-30 apb_uart_get_rxcount 函数定义.....	27
表 6-31 apb_uart_control 函数定义.....	27
表 6-32 Control Settings and Operations.....	27
表 6-33 apb_uart_get_status 函数定义.....	29
表 6-34 apb_uart_set_modem_control 函数定义.....	29
表 6-35 apb_uart_get_modem_status 函数定义.....	30
表 6-36 apb_uart_irq_handler 函数定义.....	30
表 6-37 apb_uart_dma_tx_event 函数定义.....	30
表 6-38 apb_uart_dma_rx_event 函数定义.....	31

1 关于本手册

1.1 手册内容

Gowin APB UART IP 用户指南主要包含功能特性、结构框图、功能描述、信号描述、参数描述、界面配置、编程模型、编程序列、参考设计等内容，旨在帮助用户快速了解 Gowin APB UART IP 的特性和使用方法。本手册中的软件界面截图参考的是 1.9.11.02 (64-bit) 版本，因软件版本升级，部分信息可能会略有差异，具体以用户软件版本的信息为准。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com 可以下载、查看以下相关文档：

- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS117, GW1NR 系列 FPGA 产品数据手册](#)
- [DS821, GW1NS 系列 FPGA 产品数据手册](#)
- [DS861, GW1NSR 系列 FPGA 产品数据手册](#)
- [DS841, GW1NZ 系列 FPGA 产品数据手册](#)
- [DS961, GW2ANR 系列 FPGA 产品数据手册](#)
- [DS102, GW2A 系列 FPGA 产品数据手册](#)
- [DS226, GW2AR 系列 FPGA 产品数据手册](#)
- [DS971, GW2AN-18X & 9X 器件数据手册](#)
- [DS976, GW2AN-55 器件数据手册](#)
- [DS981, GW5AT 系列 FPGA 产品数据手册](#)
- [DS1103, GW5A 系列 FPGA 产品数据手册](#)
- [DS1239, GW5AST 系列 FPGA 产品数据手册](#)
- [DS1105, GW5AS 系列 FPGA 产品数据手册](#)
- [DS1108, GW5AR 系列 FPGA 产品数据手册](#)

- [DS1118, GW5ART 系列 FPGA 产品数据手册](#)
- [SUG100, Gowin 云源软件用户指南](#)

1.3 术语、缩略语

本手册中出现的相关术语、缩略语及相关释义如表 1-1 所示。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
AMBA	Advanced Microcontroller Bus Architecture	高级微控制器总线架构
APB	Advanced Peripheral Bus	高级外围总线
DMA	Direct Memory Access	直接内存访问
FIFO	First In and First Out	先进先出队列
FPGA	Field Programmable Gate Array	现场可编程门阵列
IP	Intellectual Property	知识产权
LSB	Least Significant Byte	最低有效字节
MCR	Modem Control Register	调制解调器控制寄存器
MSB	Most Significant Byte	最高有效字节
MSR	Modem Status Register	调制解调器状态寄存器
OSCR	Over Sample Control Register	过采样控制寄存器
UART	Universal Asynchronous Receiver/Transmitter	通用异步接收器/发射器

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 概述

2.1 功能特性

Gowin APB UART IP 主要包含以下功能特性：

- 通过 AMBA 2.0 APB 接口访问寄存器
- 硬件可配置 16、32、64 和 128 字节的发送/接收 FIFO
- 可编程过采样频率（偶数倍频，范围为 8x 到 32x）
 - 编程序列兼容 16C550D UART
 - 支持每字符 5 ~ 8 位数据位
 - 支持 1、1.5 和 2 个停止位
 - 支持奇校验、偶校验和固定校验位
 - 支持 DMA 传输功能
 - 支持可编程波特率
 - 支持调制解调器控制接口
 - 支持完整的状态报告功能
 - 支持线中止、校验错误、帧错误和数据溢出检测

2.2 资源使用

通过 Verilog 语言实现 Gowin APB UART IP。因使用器件的密度、速度和等级不同，其性能和资源使用情况可能不同。以高云 GW5AT 系列 FPGA 为例，Gowin APB UART IP 资源使用情况如表 2-1 所示。

表 2-1 资源使用情况

器件系列	资源	资源使用	配置
GW5AT	Logic	677	FIFO Depth: 16 Byte
	Register	475	Same Clock Source (UCLK and PCLK): 勾选 Enable DMA: 不勾选

器件系列	资源	资源使用	配置
GW5AT	BSRAM	0	FIFO Depth: 16 Byte Same Clock Source (UCLK and PCLK): ☒ 勾选 Enable DMA: ☐ 不勾选

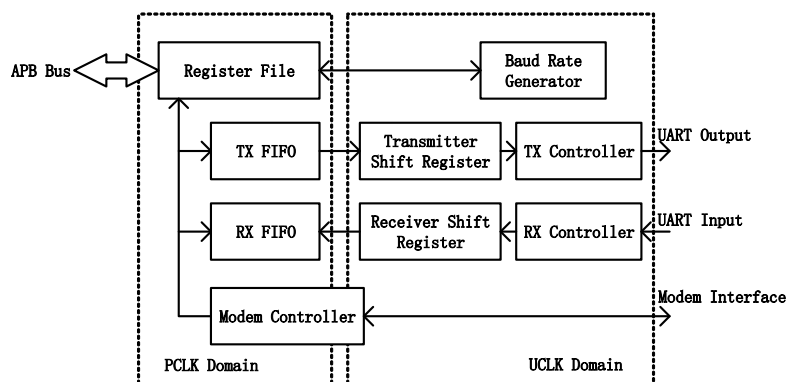
3 功能描述

Gowin APB UART IP 是一个串行通信控制器，为外部设备或调制解调器提供异步串行接口。APB UART 由一个发送器、一个接收器、一个波特率生成器、一个调制解调器控制器、一个寄存器文件组以及一个 APB 接口组成。

3.1 结构框图

Gowin APB UART IP 结构框图如图 3-1 所示。

图 3-1 结构框图



3.2 发送器

发送器包含一个发送 FIFO、一个发送移位寄存器和一个发送控制器。

发送 FIFO 用于保存要通过串行接口传输的数据。根据硬件配置和编程设置，发送 FIFO 最多可以保存 128 个字符。发送移位寄存器从发送 FIFO 中读取一个字符，用于进行下一次的传输。发送移位寄存器的功能是将并行数据转换为串行数据，将要发送的字符转换为串行位流。对于每个字符的传输，发送控制器生成一个起始位、一个可选的校验位和若干个停止位。校验位和停止位的生成可以通过线路控制寄存器进行编程。发送 FIFO 默认仅为一个单元的缓冲区，称为发送保持寄存器。只有在启用了多单元 FIFO 模式（由 FIFO 控制寄存器中的 FIFOE 位控制）时，才能工作在多存储单元模式下。

3.3 接收器

接收器包含一个接收 FIFO、一个接收移位寄存器和一个接收控制器。

接收控制器利用由波特率生成器产生的过采样时钟，在每个位传输的中间时刻进行采样。接收到的各位依次被移入接收移位寄存器，实现串行到并行的数据转换，并将转换后的字符保存到接收 FIFO 中。默认情况下，接收 FIFO 仅为一个单元的缓冲区，即接收缓冲寄存器；只有通过将 FIFO 控制寄存器中使能 FIFOE 位，才能使其工作于多单元 FIFO 模式，此模式下根据硬件配置最多可保存 128 个字符。接收控制器还会对每次数据传输检测奇偶校验错误、帧错误、数据超限错误以及线路断开等错误条件。

3.4 波特率生成器

波特率生成器以 UART 时钟（UCLK）作为源时钟，通过一个分频值进行分频。该分频值为 16 位，保存在两个独立的编程寄存器中，每个寄存器保存 8 位值。最高有效字节保存在分频锁存器最高有效字节寄存器中，最低有效字节保存在分频锁存器最低有效字节寄存器中。

采样时钟频率与波特率的比率为过采样比率，保存在过采样控制寄存器中。过采样控制寄存器的默认值为 16，通常情况下足够使用，无需进一步调整。

分频值的计算公式如下：分频值 = $UCLK / (\text{所需的波特率} \times OSCR)$

接收控制器利用过采样时钟来检测起始位的前沿，并在每个位传输的中间时刻对数据进行采样。假设过采样率为 16，当接收控制器检测到 SIN 信号的下降沿（即起始位的开始）后，会启动一个计数器，该计数器从 1 数到 16。当计数器的值等于 8 时，接收控制器采样 SIN 信号的值作为数据位；当计数器的值达到 16 时，为采样下一个位将计数器重置为 1。该过程将不断重复，直至接收到停止位。过采样时钟同样也被发送控制器用来生成其输出数据流。

过采样技术能够提供更好的时钟偏差容忍度。假设：

- T 是 APB UART 接收控制器所感知的单个位传输的周期；
- $T_{\text{transmitter}}$ 是发送器的单个位传输的周期；
- N 是一帧数据的位数，包括起始位、数据位、校验位（如有）以及停止位。

$T_{\text{transmitter}}$ 的时钟周期容差计算公式如下：

$$\left(1 - \frac{\left(0.5 - \frac{1}{OSCR}\right)}{N}\right) \times T \leq T_{\text{transmit}} \leq (N - 0.5)/(N - 1) \times T$$

由于 T 是波特率的倒数，因此该控制器生成的实际波特率与发送器的实际波特率之间的比例（即容差因子）可以在以下范围内：

$$1 - \frac{\left(0.5 - \frac{1}{OSCR}\right)}{N} \leq \frac{\text{Actual Baud Rate}}{\text{Actual Transmitter Baud Rate}} \leq (N - 0.5)/(N - 1)$$

如果一个字符包含 1 个起始位、8 个数据位、1 个校验位以及 1 个停止位，那么 $N=11$ ($1+8+1+1$)，此时容差因子的范围为 0.9602 至 1.05。在典型 N 和 $OSCR$ 值下，实际发送波特率的时钟容差因子（以百分比表示）如表 3-1 所示。

表 3-1 时钟变化容差因子

OSCR	N = 9	N = 10	N = 11	N = 12
8	95.83% - 106.25%	96.25% - 105.56%	96.59% - 105.00%	96.88% - 104.55%
16	95.14% - 106.25%	95.63% - 105.56%	96.02% - 105.00%	96.35% - 104.55%
32	94.79% - 106.25%	95.31% - 105.56%	95.74% - 105.00%	96.09% - 104.55%

3.5 调制解调器控制器

调制解调器控制器提供了调制解调器控制功能，此外，还提供了自动流控功能，以减少软件管理的工作量。

Gowin APB UART IP 的流控可以通过 RTS/CTS 握手实现。没有流控时，当数据传输速率超过数据接收速率时，可能会发生溢出错误。流控保证只有当接收器有足够空间接收数据时，数据传输才会继续进行。

Gowin APB UART IP 的自动流控功能包括自动 RTS 和自动 CTS，前者用于传入数据，而后者用于传出数据。

对于自动 RTS，控制器的 RTS 输出应连接到 APB UART 连接的另一端的 CTS 输入。

- N 是接收 FIFO 触发电平的阈值。
- B_n 是接收 FIFO 接收到的字节数。
- 当 $B_n \geq N$ 时，RTS 被取消置位。
- 当 $B_n < N$ 时，RTS 自动使能，请求 APB UART 连接的另一端发送更多数据。

对于自动 CTS，控制器的 CTS 输入连接到 APB UART 连接的另一端的 RTS 输出。

APB UART 控制器在发送下一个字符之前会等待 CTS 的使能。要阻止控制器发送下一个字符，必须在当前字符的停止位之前取消置位 CTS。

3.6 回环模式

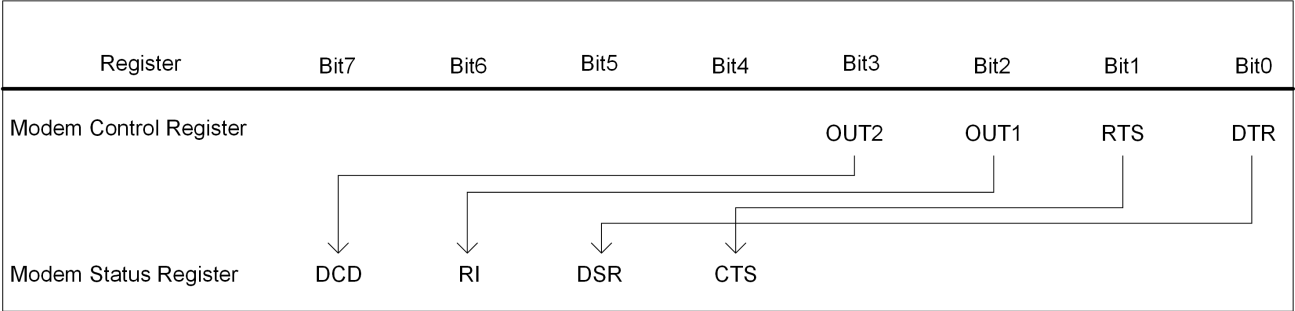
APB UART 提供回环模式，用于在不连接外部设备的情况下进行诊断测试。当启用回环模式时，控制器的行为如下：

- 输出信号（uart_sout、modem_rstn、modem_dtrn、uart_out1n、uart_out2n）与发送控制器断开连接，并驱动高电平来避免在连接存在的情况下干扰串行连接的另一端。
- 输入信号（uart_sin、modem_ctsn、modem_dsrn、modem_rin、modem_dcsn）与接收控制器断开连接，并被忽略。
- 发送控制器原本用于输出到 uart_sout 信号的值，通过内部路由取代 uart_sin 的输入信号，供接收控制器使用，因此发送控制器的每一位数据都会被回传，并由接收控制器接收。

- 调制解调器控制器原本用于输出到 modem_rstn、modem_dtrn、uart_out1n 和 uart_out2n 信号的值，通过内部路由取代 modem_ctsn、modem_dsrn、modem_rin 和 modem_dcsn 的输入信号，供调试解调器控制器使用。

写入发送保持寄存器的值会被接收缓冲寄存器内部接收，写入调制解调器控制寄存器（bit3 ~ bit0）的值会被路由到调制解调器状态寄存器（bit7 ~ bit4）。在回环模式下，整个位传输路径都会被测试，只有输入/输出端口被隔离在诊断活动之外。调制解调器控制寄存器中的位字段与调制解调器状态寄存器中的位字段之间的关系，如图 3-2 所示。

图 3-2 回环模式中 MCR 和 MSR 的关系



3.7 DMA 操作

Gowin APB UART IP 控制器提供两对硬件握手信号，用于与 DMA 控制器协作进行数据传输，一对用于数据接收，另一对用于数据发送。

当接收 FIFO 达到 FIFO 控制寄存器的接收 FIFO 触发级别（RFIFOT）字段设置的阈值时，APB UART 控制器将断言 dma_rx_req 信号来请求数据传输。DMA 控制器随后从接收 FIFO 传输一个字符，并断言 dma_rx_ack 信号。接下来，APB UART 控制器将去断言 dma_rx_req 信号，而 DMA 控制器将去断言 dma_rx_ack 信号。除非接收 FIFO 为空，否则 APB UART 控制器将再次断言 dma_rx_req 信号。

当发送 FIFO 达到 FIFO 控制寄存器的发送 FIFO 触发级别（TFIFOT）字段设置的阈值时，APB UART 控制器将断言 dma_tx_req 信号来请求数据传输。DMA 控制器随后向发送 FIFO 传输一个字符，并断言 dma_tx_ack 信号。接下来，APB UART 控制器将去断言 dma_tx_req 信号，而 DMA 控制器将去断言 dma_tx_ack 信号。除非发送 FIFO 已满，否则 APB UART 控制器将再次断言 dma_tx_req 信号。

4 信号描述

Gowin APB UART IP 的信号描述如表 4-1 所示。

表 4-1 信号描述

Signal Name	I/O	Width	Description
uclk	input	1	UART clock source
urstn	input	1	UART system reset; active low
dma_rx_ack	input	1	DMA RX acknowledgement
dma_rx_req	output	1	DMA RX request
dma_tx_ack	input	1	DMA TX acknowledgement
dma_tx_req	output	1	DMA TX request
paddr	input	[5:2]	APB address bus
pclk	input	1	APB clock, used to time all the bus transfers
penable	input	1	APB enable signal
presetn	input	1	APB reset signal; active low
psel	input	1	APB select signal When set to 1, this signal indicates that the slave device has been selected by the APB bridge and that a data transfer is required.
pwdata	input	[31:0]	APB write data bus
pwrite	input	1	APB transfer direction signal This signal indicates a write access when high and a read access when low.
prdata	output	[31:0]	APB read data bus
pready	output	1	APB ready signal
pslverr	output	1	APB slave error signal
uart_ctsn	input	1	Modem clear to send (CTS); active low

Signal Name	I/O	Width	Description
uart_dcdn	input	1	Modem data carrier detect; active low
uart_dsrn	input	1	Modem data set ready; active low
uart_rin	input	1	Modem ring indicator; active low
uart_sin	input	1	UART serial input data When there is no communication, uart_sin stays at 1.
uart_dtrn	output	1	Modem data terminal ready; active low
uart_intr	output	1	UART interrupt signal
uart_out1n	output	1	UART user-defined output 1; active low
uart_out2n	output	1	UART user-defined output 2; active low
uart_rtsn	output	1	Modem request to send (RTS); active low
uart_sout	output	1	UART serial output data When there is no communication, uart_sout stays at 1.

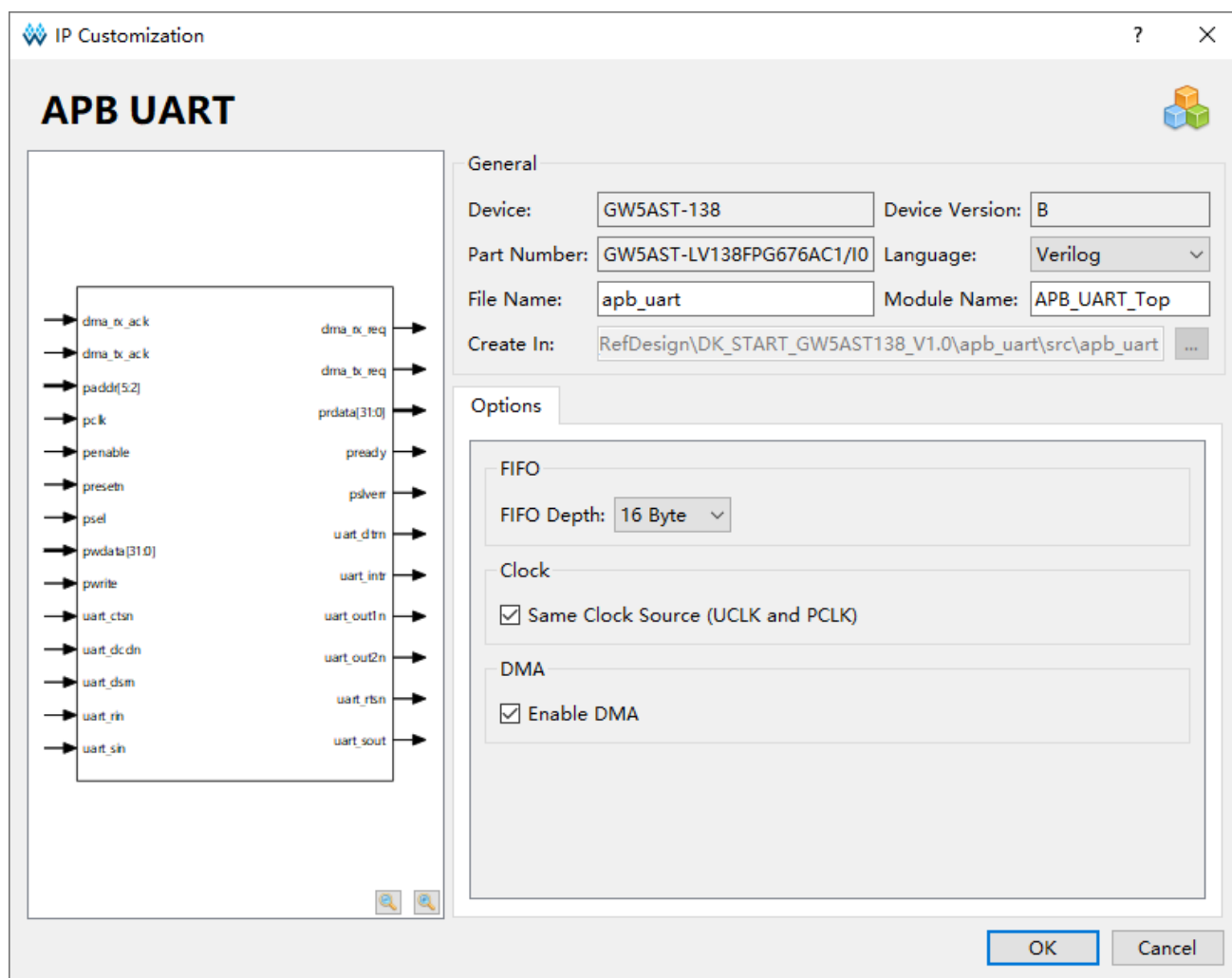
5 界面配置

用户可以在高云半导体云源软件的 IP Core Generator 工具，调用并配置 Gowin APB UART IP。

选择菜单栏“Tool > IP Core Generator”或工具栏“”，打开 IP Core Generator，IP 列表中选择“Soft IP Core > Microprocessor System > Peripheral > APB UART 1.0”。

Gowin APB UART IP 界面配置如[图 5-1](#) 所示。

图 5-1 界面配置



选项配置描述如下所示：

- **FIFO Depth:** 发送 FIFO 和接收 FIFO 的深度，可以配置为 16、32、64 和 128 Byte，默认值为 16 Byte。
- **Same Clock Source (UCLK and PCLK):** Gowin APB UART 的 UCLK 与 PCLK 是否具有相同的时钟源，如果选择该选项，UCLK 使用 PCLK 时钟，默认勾选。
- **Enable DMA:** Gowin APB UART 是否支持 DMA 功能。如果选择了 Enable DMA，则 APB UART 支持 DMA 功能，默认不勾选。

6 编程模型

6.1 寄存器

6.1.1 寄存器概述

Gowin APB UART IP 的寄存器定义如表 6-1 所示。Gowin APB UART IP 寄存器定义位于...\\lib\\driver\\apb_uart.h。

表 6-1 寄存器定义

地址偏移	名称	描述
0x00	IDREV	ID 和修订寄存器
0x04~0x0C	-	保留
0x10	CFG	硬件配置寄存器
0x14	OSCR	过采样控制寄存器
0x18~0x1C	-	保留
0x20	DLAB = 0	
	RBR	接收缓冲寄存器（只读）
	THR	发送保持寄存器（只写）
	DLAB = 1	
	DLL	分频锁存器最低有效字节
0x24	DLAB = 0	
	IER	中断使能寄存器
	DLAB = 1	
	DLM	分频锁存器最高有效字节
0x28	IIR	中断识别寄存器（只读）
	FCR	FIFO 控制寄存器（只写）
0x2C	LCR	线路控制寄存器

地址偏移	名称	描述
0x30	MCR	调制解调器控制寄存器
0x34	LSR	线路状态寄存器
0x38	MSR	调制解调器状态寄存器
0x3C	SCR	暂存寄存器

以下各节详细描述 Gowin APB UART IP 寄存器定义。

寄存器类型缩略语概括如下：

- RO: Read-only
- WO: Write-only
- R/W: Readable and writable
- W1C: Write 1 to clear
- RC: Read clear

6.1.2 ID 和修订寄存器（0x00）

ID 和修订寄存器用于保存 ID 和修订编号，初始值依赖于所用版本。ID 和修订寄存器定义如表 6-2 所示。

表 6-2 ID 和修订寄存器

Name	Bit	Type	Description	Reset
ID	31:16	RO	ID number for UART	0x0001
Major	15:4	RO	Major revision number	Revision dependent
Minor	3:0	RO	Minor revision number	Revision dependent

6.1.3 硬件配置寄存器（0x10）

硬件配置寄存器定义如表 6-3 所示。

表 6-3 硬件配置寄存器

Name	Bit	Type	Description	Reset
-	31:2	-	Reserved	0x00
FIFO_DEPTH	1:0	RO	The depth of RXFIFO and TXFIFO 0: 16-byte FIFO 1: 32-byte FIFO 2: 64-byte FIFO 3: 128-byte FIFO	Configuration dependent

6.1.4 过采样控制寄存器（0x14）

过采样控制寄存器定义如表 6-4 所示。

表 6-4 过采样控制寄存器

Name	Bit	Type	Description	Reset
-	31:5	-	Reserved	0x0
OSC	4:0	R/W	Over-sample control The value must be an even number; any odd value writes to this field will be converted to an even value. OSC = 0: The over-sample ratio is 32 OSC ≤ 8: The over-sample ratio is 8 8 < OSC < 32: The over-sample ratio is OSC	0x10

6.1.5 接收缓冲寄存器（当 DLAB = 0）（0x20）

接收缓冲寄存器包括两种模式，FIFO 模式和 BUFFER 模式。FIFO 控制寄存器的 Bit0 位（FIFOE）控制这两种模式的选择。当 FIFOE 为 1 时（FIFO 模式），接收缓冲寄存器为 RXFIFO；当 FIFOE 为 0 时（BUFFER 模式），接收缓冲寄存器为一个字节缓存区。接收缓冲寄存器定义如表 6-5 所示。

表 6-5 接收缓冲寄存器（当 DLAB = 0）

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
RBR	7:0	RO	Receive data read port	0x0

6.1.6 发送保持寄存器（当 DLAB = 0）（0x20）

发送保持寄存器包括两种模式，FIFO 模式和 BUFFER 模式。FIFO 控制寄存器的 Bit0 位（FIFOE）控制这两种模式的选择。当 FIFOE 为 1 时（FIFO 模式），发送保持寄存器为 TXFIFO；当 FIFOE 为 0 时（BUFFER 模式），发送保持寄存器为一个字节缓存区。发送保持寄存器定义如表 6-6 所示。

表 6-6 发送保持寄存器（当 DLAB = 0）

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
THR	7:0	WO	Transmit data write port	0x0

6.1.7 中断使能寄存器（当 DLAB = 0）（0x24）

中断使能寄存器定义如表 6-7 所示。

表 6-7 中断使能寄存器（当 DLAB = 0）

Name	Bit	Type	Description	Reset
-	31:4	-	Reserved	0x0
EMSI	3	R/W	Enable modem status interrupt	0x0

Name	Bit	Type	Description	Reset
EMSI	3	R/W	The interrupt asserts when the status of one of the following occurs: The status of modem_rin, modem_dcdn, modem_dsrn or modem_ctsn (If the auto-cts mode is disabled) has been changed. If the auto-cts mode is enable (MCR bit4 (AFE) = 1), modem_ctsn would be used to control the transmitter.	0x0
ELSI	2	R/W	Enable receiver line status interrupt	0x0
ETHEI	1	R/W	Enable transmitter holding register interrupt	0x0
ERBI	0	R/W	Enable received data available interrupt and the character timeout interrupt 0: Disable 1: Enable	0x0

6.1.8 分频锁存器最低有效字节（当 DLAB = 1）（0x20）

分频锁存器最低有效字节保存分频值，用于从 APB UART 时钟源（UCLK）产生采样时钟。分频锁存器最低有效字节的大小为 16 位（2 个字节），寄存器保存分频锁存器的最低有效字节。分频锁存器最低有效字节的有效值在 1 和 65535（ $2^{16}-1$ ）之间，包括 1 和 65535。分频锁存器最低有效字节定义如表 6-8 所示。

表 6-8 分频锁存器最低有效字节（当 DLAB = 1）

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
DLL	7:0	R/W	Least significant byte of the Divisor Latch	0x1

6.1.9 分频锁存器最高有效字节（当 DLAB = 1）（0x24）

分频锁存器最高有效字节保存分频值，用于从 UART 时钟源（UCLK）产生采样时钟。分频锁存器最高有效字节的大小为 16 位（2 个字节），寄存器保存分频锁存器的最高有效字节。分频锁存器最高有效字节的有效值在 1 和 65535（ $2^{16}-1$ ）之间，包括 1 和 65535。分频锁存器最高有效字节定义如表 6-9 所示。

表 6-9 分频锁存器最高有效字节（当 DLAB = 1）

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
DLM	7:0	R/W	Most significant byte of the Divisor Latch	0x0

6.1.10 中断识别寄存器（0x28）

中断识别寄存器定义如表 6-10 所示，中断控制类型如表 6-11 所示。

表 6-10 中断识别寄存器

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
FIFOED	7:6	RO	FIFOs enabled These two bits are 1 when bit0 of the FIFO Control Register (FIFOE) is set to 1	0x0
-	5:4	-	Reserved	0x0
INTRID	3:0	RO	Interrupt ID See table 6-11 for encodings	0x1

表 6-11 中断控制类型

Interrupt Identification Register					Interrupt Type	Interrupt Source Description	Interrupt Reset Method
Bit3	Bit2	Bit1	Bit0	Priority Level			
0	0	0	1	None	None	None	None
0	1	1	0	1	Receiver Line status	Overrun errors, parity errors, framing errors, or line breaks	Read the Line Status Register (LSR)
0	1	0	0	2	Received data available	If FIFOE is disabled, there is one received data available in the RBR. If FIFOE is enabled, the numbers of received data available reach the trigger level (RFIFOT). The interrupt signal will stay active until the number of data available becoming smaller than the trigger level.	Read the Receiver Buffer Register (RBR)
1	1	0	0	2	Character timeout	When FIFOE is enabled and no character has been removed from or input to receive FIFO and there is at least one character in receive FIFO during the last four character times.	Read the Receiver Buffer Register (RBR)
0	0	1	0	3	Transmitter Holding Register empty	If FIFOE is disabled, the 1-byte THR is empty.	Write the Transmitter Holding Register (THR) or Read the Interrupt

Interrupt Identification Register					Interrupt Type	Interrupt Source Description	Interrupt Reset Method
Bit3	Bit2	Bit1	Bit0	Priority Level			
0	0	1	0	3	Transmitter Holding Register empty	If FIFOE is enabled, the whole 16-byte transmit FIFO is empty.	Identification Register (IIR).
0	0	0	0	4	Modem status	The Modem Status Register (MSR) bit [3:0] is not 0. One of the following events occurred: Clear To Send (CTS), Data Set Ready (DSR), Ring Indicator (RI), or Data Carrier Detect (DCD)	Read the Modem Status Register (MSR)

6.1.11 FIFO 控制寄存器 (0x28)

FIFO 控制寄存器定义如表 6-12 所示。

表 6-12 FIFO 控制寄存器

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
RFIFOT	7:6	WO	Receiver FIFO trigger level Please refer to table 6-13	0x0
TFIFOT	5:4	WO	Transmitter FIFO trigger level Please refer to table 6-14	0x0
DMAE	3	WO	DMA enable 0: Disable 1: Enable	0x0
TFIFORST	2	WO	Transmitter FIFO reset Write 1 to clear all bytes in the TXFIFO and resets its counter. The Transmitter Shift Register is not cleared. This bit will automatically be cleared.	0x0
RFIFORST	1	WO	Receiver FIFO reset Write 1 to clear all bytes in the RXFIFO and resets its counter. The Receiver Shift Register is not cleared. This bit will automatically be cleared.	0x0
FIFOE	0	WO	FIFO enable Write 1 to enable both the transmitter and receiver FIFOs. The FIFOs are reset when the value of this bit toggles.	0x0

接收 FIFO 触发级别定义如表 6-13 所示。

表 6-13 接收 FIFO 触发级别

RFIFOT Value	RXFIFO Trigger Level			
	16-byte RXFIFO	32-byte RXFIFO	64-byte RXFIFO	128-byte RXFIFO
0	Not empty	Not empty	Not empty	Not empty
1	More than 3	More than 7	More than 15	More than 31
2	More than 7	More than 15	More than 31	More than 63
3	More than 13	More than 27	More than 55	More than 111

发送 FIFO 触发级别定义如表 6-14 所示。

表 6-14 发送 FIFO 触发级别

TFIFOT Value	TXFIFO Trigger Level			
	16-byte TXFIFO	32-byte TXFIFO	64-byte TXFIFO	128-byte TXFIFO
0	Not full	Not full	Not full	Not full
1	Less than 12	Less than 24	Less than 48	Less than 96
2	Less than 8	Less than 16	Less than 32	Less than 64
3	Less than 4	Less than 8	Less than 16	Less than 32

6.1.12 线路控制寄存器（0x2C）

线路控制寄存器定义如表 6-15 所示。

表 6-15 线路控制寄存器

Name	Bit	Type	Description	Reset
-	[31:8]	-	Reserved	0x0
DLAB	7	R/W	Divisor latch access bit	0x0
BC	6	R/W	Break control	0x0
SPS	5	R/W	Stick parity 1: Parity bit is constant 0 or 1, depending on bit4 (EPS). 0: Disable the sticky bit parity. Please refer to table 6-16.	0x0
EPS	4	R/W	Even parity select 1: Even parity (an even number of logic-1 is in the data and parity bits). 0: Odd parity. Please refer to table 6-16.	0x0
PEN	3	R/W	Parity enable	0x0

Name	Bit	Type	Description	Reset
PEN	3	R/W	When this bit is set, a parity bit is generated in transmitted data before the first STOP bit and the parity bit would be checked for the received data. Please refer to table 6-16.	0x0
STB	2	R/W	Number of STOP bits 0: 1 bits 1: The number of STOP bit is based on the WLS setting When WLS = 0, STOP bit is 1.5 bits When WLS = 1, 2, 3, STOP bit is 2 bits	0x0
WLS	1:0	R/W	Word length setting 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits	0x0

奇偶校验位选择定义如表 6-16 所示。

表 6-16 奇偶校验位选择

PEN (bit3)	SPS (bit5)	EPS (bit4)	Parity Bit
0	X	X	No parity bit
1	0	0	Parity is odd
1	0	1	Parity is even
1	1	01	Parity bit is always 1
1	1		Parity bit is always 0

6.1.13 调制解调器控制寄存器（0x30）

调制解调器控制寄存器，控制调制解调器状态信号输出，以及回环模式和自动流控制。
调制解调器控制寄存器定义如表 6-17 所示。

表 6-17 调制解调器控制寄存器

Name	Bit	Type	Description	Reset
-	31:6	-	Reserved	0x0
AFE	5	R/W	Auto flow control enable 0: Disable 1: The auto-CTS and auto-RTS setting is based on the RTS bit setting: When RTS = 0, auto-CTS only When RTS = 1, auto-CTS and auto-RTS	0x0

Name	Bit	Type	Description	Reset
LOOP	4	R/W	Enable loopback mode 0: Disable 1: Enable	0x0
OUT2	3	R/W	User-defined output 2 This bit controls the uart_out2n output. 0: The uart_out2n output signal will be driven HIGH 1: The uart_out2n output signal will be driven LOW	0x0
OUT1	2	R/W	User-defined output 1 This bit controls the uart_out1n output. 0: The uart_out1n output signal will be driven HIGH 1: The uart_out1n output signal will be driven LOW	0x0
RTS	1	R/W	Request to send This bit controls the modem_rstn output 0: The modem_rstn output signal will be driven HIGH 1: The modem_rstn output signal will be driven LOW	0x0
DTR	0	R/W	Data terminal ready This bit controls the modem_dtrn output. 0: The modem_dtrn output signal will be driven HIGH 1: The modem_dtrn output signal will be driven LOW	0x0

6.1.14 线路状态寄存器（0x34）

线路状态寄存器，报告发送器和接收器的状态。

线路状态寄存器定义如表 6-18 所示。

表 6-18 线路状态寄存器

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
ERRF	7	RO	Error in RXFIFO In the FIFO mode, this bit is set when there is at least one parity error, framing error, or line break associated with data in the RXFIFO. It is cleared when this register is read and there is no more error for the rest of data in the RXFIFO	0x0
TEMT	6	RO	Transmitter empty This bit is 1 when the THR (TXFIFO in the FIFO mode) and the Transmitter Shift Register (TSR) are both empty. Otherwise, it is zero.	0x1
THRE	5	RO	Transmitter Holding Register empty	0x1

Name	Bit	Type	Description	Reset
THRE	5	RO	This bit is 1 when the THR (TXFIFO in the FIFO mode) is empty. Otherwise, it is zero. If the THRE interrupt is enabled, an interrupt is triggered when THRE becomes 1.	0x1
LBreak	4	RO	Line break This bit is set when the uart_sin input signal was held LOW for longer than the time for a full-word transmission. A full-word transmission is the transmission of the START, data, parity, and STOP bits. It is cleared when this register is read. In the FIFO mode, this bit indicates the line break for the received data at the top of the RXFIFO.	0x0
FE	3	RO	Framing error This bit is set when the received STOP bit is not HIGH. It is cleared when this register is read. In the FIFO mode, this bit indicates the framing error for the received data at the top of the RXFIFO.	0x0
PE	2	RO	Parity error This bit is set when the received parity does not match with the parity selected in the LCR [5:4]. It is cleared when this register is read. In the FIFO mode, this bit indicates the parity error for the received data at the top of the RXFIFO.	0x0
OE	1	RO	Overrun error This bit indicates that data in the Receiver Buffer Register (RBR) is overrun.	0x0
DR	0	RO	Data ready This bit is set when there are incoming received data in the Receiver Buffer Register (RBR). It is cleared when all of the received data are read.	0x0

6.1.15 调制解调器状态寄存器（0x38）

调制解调器状态寄存器定义如表 6-19 所示。

表 6-19 调制解调器状态寄存器

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
DCD	7	RO	Data carrier detect 0: The modem_dcdn input signal is HIGH.	0x0

Name	Bit	Type	Description	Reset
DCD	7	RO	1: The modem_dcdn input signal is LOW.	0x0
RI	6	RO	Ring indicator 0: The modem_rin input signal is HIGH. 1: The modem_rin input signal is LOW.	0x0
DSR	5	RO	Data set ready 0: The modem_dsrn input signal is HIGH. 1: The modem_dsrn input signal is LOW.	0x0
CTS	4	RO	Clear to send 0: The modem_ctsn signal is HIGH. 1: The modem_ctsn signal is LOW.	0x0
DDCD	3	RC	Delta data carrier detect This bit is set when the state of the modem_dcdn input signal has been changed since the last time this register is read. Otherwise, it is zero.	0x0
TERI	2	RC	Trailing edge ring indicator This bit is set when the state of the modem_rin input signal has been changed from LOW to HIGH since the last time this register is read.	0x0
DDSR	1	RC	Delta data set ready This bit is set when the state of the modem_dsrn input signal has been changed since the last time this register is read.	0x0
DCTS	0	RC	Delta clear to send This bit is set when the state of the modem_ctsn input signal has been changed since the last time this register is read.	0x0

6.1.16 暂存寄存器（0x3C）

暂存寄存器定义如表 6-20 所示。

表 6-20 暂存寄存器

Name	Bit	Type	Description	Reset
-	31:8	-	Reserved	0x0
SCR	7:0	R/W	An one-byte storage register with no UART related function; available to software with no usage restrictions.	0x0

6.2 驱动函数

6.2.1 驱动函数概述

Gowin APB UART IP 驱动函数定义如表 6-21 所示。Gowin APB UART IP 驱动函数定义位于...\lib\driver\apb_uart_driver.h 和 apb_uart_driver.c。

表 6-21 驱动函数定义

驱动函数	描述
apb_uart_get_capabilities	获取 APB UART 驱动的功能信息
apb_uart_initialize	初始化 APB UART 接口
apb_uart_uninitialize	卸载 APB UART 接口
apb_uart_power_control	指定 APB UART 接口的功耗模式
apb_uart_send	发送数据到 APB UART 发送器
apb_uart_receive	从 APB UART 接收器接收数据
apb_uart_transfer	通过 APB UART 接口传输数据
apb_uart_get_txcount	获取 APB UART 接口发送数据的数量
apb_uart_get_rxcount	获取 APB UART 接口接收数据的数量
apb_uart_control	配置 APB UART 接口的设置，执行指定的操作
apb_uart_get_status	获取 APB UART 接口的状态
apb_uart_set_modem_control	控制 APB UART 接口的调制解调器控制线的状态
apb_uart_get_modem_status	获取 APB UART 接口的调制解调器控制线的状态
apb_irq_handler	APB UART 中断处理程序
apb_uart_dma_tx_event	APB UART DMA 发送事件
apb_uart_dma_rx_event	APB UART DMA 接收事件

以下各节详细描述 APB UART 的驱动函数定义。

6.2.2 apb_uart_get_capabilities

apb_uart_get_capabilities 函数定义如表 6-22 所示。

表 6-22 apb_uart_get_capabilities 函数定义

原型	APB_UART_CAPABILITIES apb_uart_get_capabilities (APB_UART_RESOURCES *apb_uartx)
描述	获取 APB UART 驱动的功能信息
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	APB UART 驱动的功能信息

6.2.3 apb_uart_initialize

apb_uart_initialize 函数定义如表 6-23 所示。

表 6-23 apb_uart_initialize 函数定义

原型	int apb_uart_initialize (APB_UART_SignalEvent_t cb_event, APB_UART_RESOURCES *apb_uartx)
描述	初始化 APB UART 接口
参数	cb_event: 指向 APB_UART_SignalEvent 回调函数的指针 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.4 apb_uart_uninitialize

apb_uart_uninitialize 函数定义如表 6-24 所示。

表 6-24 apb_uart_uninitialize 函数定义

原型	int apb_uart_uninitialize (APB_UART_RESOURCES *apb_uartx)
描述	卸载 APB UART 接口
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.5 apb_uart_power_control

apb_uart_power_control 函数定义如表 6-25 所示。

表 6-25 apb_uart_power_control 函数定义

原型	int apb_uart_power_control (APB_UART_POWER_STATE state, APB_UART_RESOURCES *apb_uartx)
描述	指定 APB UART 接口的功耗模式
参数	state: APB UART 接口功耗模式, 包括: • APB_UART_POWER_FULL: to set up peripherals for data transfers, enable interrupts and DMA • APB_UART_POWER_LOW: to enable power-saving • APB_UART_POWER_OFF: to terminate pending data transfers and disable peripherals, related interrupts and DMA apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.6 apb_uart_send

apb_uart_send 函数定义如表 6-26 所示。

表 6-26 apb_uart_send 函数定义

原型	int apb_uart_send (const void *data, unsigned int num, APB_UART_RESOURCES *apb_uartx)
----	---

描述	发送数据到 APB UART 发送器
参数	data: 指向发送数据缓存区的指针 num: 发送数据的长度 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.7 apb_uart_receive

apb_uart_receive 函数定义如表 6-27 所示。

表 6-27 apb_uart_receive 函数定义

原型	int apb_uart_receive (void *data, unsigned int num, APB_UART_RESOURCES *apb_uartx)
描述	从 APB UART 接收器接收数据
参数	data: 指向接收数据缓存区的指针 num: 接收数据的长度 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.8 apb_uart_transfer

apb_uart_transfer 函数定义如表 6-28 所示。

表 6-28 apb_uart_transfer 函数定义

原型	int apb_uart_transfer (const void *data_out, void *data_in, unsigned int num, APB_UART_RESOURCES *apb_uartx)
描述	通过 APB UART 接口传输数据
参数	data_out: 指向发送数据缓存区的指针 data_in: 指向接收数据缓存区的指针 num: 传输数据的长度 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.9 apb_uart_get_txcount

apb_uart_get_txcount 函数定义如表 6-29 所示。

表 6-29 apb_uart_get_txcount 函数定义

原型	unsigned int apb_uart_get_txcount (APB_UART_RESOURCES *apb_uartx)
描述	获取 APB UART 接口发送数据的数量
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	APB UART 最后一次执行发送传输时, 发送数据的数量

6.2.10 apb_uart_get_rxcount

apb_uart_get_rxcount 函数定义如表 6-30 所示。

表 6-30 apb_uart_get_rxcount 函数定义

原型	unsigned int apb_uart_get_rxcount (APB_UART_RESOURCES *apb_uartx)
描述	获取 APB UART 接口接收数据的数量
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	APB UART 最后一次执行接收传输时，接收数据的数量

6.2.11 apb_uart_control

apb_uart_control 函数定义如表 6-31 所示。

表 6-31 apb_uart_control 函数定义

原型	int apb_uart_control (unsigned int control, unsigned int arg, APB_UART_RESOURCES *apb_uartx)
描述	配置 APB UART 接口的设置，执行指定的操作
参数	control: APB UART 驱动接口的一种设置或执行的一种操作 arg: 指定设置或操作的附加信息 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误，返回一个负值

“control”和“arg”设置与操作如表 6-32 所示。

表 6-32 Control Settings and Operations

Options for control	arg specifies	Settings or operations
Operation modes (Bits: 0~7)		
APB_UART_MODE_ASYNCHRONOUS	baudrate	Sets to be asynchronous UART mode
APB_UART_MODE_SYNCHRONOUS_MASTER	baudrate	Sets to the synchronous master mode with clock signal generation
APB_UART_MODE_SYNCHRONOUS_SLAVE		Sets to the synchronous slave mode with external clock signal
APB_UART_MODE_SINGLE_WIRE	baudrate	Sets to the single-wire (half-duplex) mode
APB_UART_MODE_IRDA	baudrate	Sets to the infra-red data mode
APB_UART_MODE_SMART_CARD	baudrate	Sets to the Smart Card mode
Data bit (Bits: 8~11)		

Options for control	arg specifies	Settings or operations
APB_UART_DATA_BITS_5		Sets to 5 data bits
APB_UART_DATA_BITS_6		Sets to 6 data bits
APB_UART_DATA_BITS_7		Sets to 7 data bits
APB_UART_DATA_BITS_8		Sets to 8 data bits
APB_UART_DATA_BITS_9		Sets to 9 data bits
Parity bit (Bits: 12~13)		
APB_UART_PARITY_NONE		Sets to no parity (default)
APB_UART_PARITY_EVEN		Sets to even parity
APB_UART_PARITY_ODD		Sets to odd parity
Stop bit (Bits: 14~15)		
APB_UART_STOP_BITS_1		Sets to 1 stop bit (default)
APB_UART_STOP_BITS_2		Sets to 2 stop bits
APB_UART_STOP_BITS_1_5		Sets to 1.5 stop bits
APB_UART_STOP_BITS_0_5		Sets to 0.5 stop bits
Flow control (Bits: 16~17)		
APB_UART_FLOW_CONTROL_NONE		Sets to have no flow control signal (default)
APB_UART_FLOW_CONTROL_RTS		Sets to use the RTS flow control signal
APB_UART_FLOW_CONTROL_CTS		Sets to use the CTS flow control signal
APB_UART_FLOW_CONTROL_RTS_CTS		Sets to use the RTS and CTS flow control signal
Clock parity (Bit: 18)		
APB_UART_CPOL0		CPOL=0 (default): Data are captured on the rising edge
APB_UART_CPOL1		CPOL=1: Data are captured on the falling edge
Clock phase (Bit: 19)		
APB_UART_CPHA0		CPHA=0 (default): Data are sampled on the first edge
APB_UART_CPHA1		CPHA=1:

Options for control	arg specifies	Settings or operations
APB_UART_CPHA1		Data are sampled on the second edge
Other operations (Bits: 0~19)		
APB_UART_SET_DEFAULT_TX_VALUE	transmit value	Sets the default transmit value
APB_UART_SET_IRDA_PULSE	0=3/16 of bit period	Sets the IrDA pulse value in ns
APB_UART_SET_SMART_CARD_GUARD_TIME	number of bit periods	Sets the Smart Card guard time
APB_UART_SET_SMART_CARD_CLOCK	0=Clock not set	Sets the Smart Card clock in Hz
APB_UART_CONTROL_SMART_CARD_NACK	0=disable; 1=enable	Enables or disables the Smart Card NACK generation
APB_UART_CONTROL_TX	0=disable; 1=enable	Enables or disables the transmitter
APB_UART_CONTROL_RX	0=disable; 1=enable	Enables or disables the receiver
APB_UART_CONTROL_BREAK	0=disable; 1=enable	Enables or disables continuous break transmission
APB_UART_ABORT_SEND		Aborts the send operations
APB_UART_ABORT_RECEIVE		Aborts the receive operation
APB_UART_ABORT_TRANSFER		Aborts the transfer operation

6.2.12 apb_uart_get_status

apb_uart_get_status 函数定义如表 6-33 所示。

表 6-33 apb_uart_get_status 函数定义

原型	APB_UART_STATUS apb_uart_get_status (APB_UART_RESOURCES *apb_uartx)
描述	获取 APB UART 接口的状态
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	APB UART 接口的当前状态

6.2.13 apb_uart_set_modem_control

apb_uart_set_modem_control 函数定义如表 6-34 所示。

表 6-34 apb_uart_set_modem_control 函数定义

原型	int apb_uart_set_modem_control (APB_UART_MODEM_CONTROL control, APB_UART_RESOURCES *apb_uartx)
----	--

描述	控制 APB UART 接口的调制解调器控制线的状态
参数	control: 启动或释放调制解调器的控制线, 可用选项包括: • APB_UART_RTS_CLEAR: to deactivate RTS • APB_UART_RTS_SET: to activate RTS • APB_UART_DTR_CLEAR: to deactivate DTR • APB_UART_DTR_SET: to activate DTR apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	如果发生执行错误, 返回一个负值。

6.2.14 apb_get_modem_status

apb_uart_get_modem_status 函数定义如表 6-35 所示。

表 6-35 apb_uart_get_modem_status 函数定义

原型	int apb_uart_set_modem_control (APB_UART_MODEM_CONTROL control, APB_UART_RESOURCES *apb_uartx)
描述	获取 APB UART 接口的调制解调器控制线的状态
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	APB UART 接口的调制解调器控制线的当前状态

6.2.15 apb_uart_irq_handler

apb_uart_irq_handler 函数定义如表 6-36 所示。

表 6-36 apb_uart_irq_handler 函数定义

原型	void apb_uart_irq_handler (APB_UART_RESOURCES *apb_uartx)
描述	APB UART 中断处理程序
参数	apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	无

6.2.16 apb_uart_dma_tx_event

apb_uart_dma_tx_event 函数定义如表 6-37 所示。

表 6-37 apb_uart_dma_tx_event 函数定义

原型	void apb_uart_dma_tx_event (unsigned int event, APB_UART_RESOURCES *apb_uartx)
描述	APB UART DMA TX 事件
参数	event: DMA TX 事件 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	无

6.2.17 apb_uart_dma_rx_event

apb_uart_dma_rx_event 函数定义如表 6-38 所示。

表 6-38 apb_uart_dma_rx_event 函数定义

原型	void apb_uart_dma_rx_event (unsigned int event, APB_UART_RESOURCES *apb_uartx)
描述	APB UART DMA RX 事件
参数	event: DMA RX 事件 apb_uartx: 指向 APB_UART_RESOURCES 结构体的指针
返回值	无

7 编程序列

7.1 设置

设置 APB UART 的编程步骤如下所示：

- 波特率：38400
- 字长：8 位
- 校验位：无
- 停止位数：1
- 流控：无

步骤 1 设置波特率。

1. 设置线路控制寄存器的 DLAB 位为 1。
2. 设置分频锁存器最低有效字节和分频锁存器最高有效字节寄存器，选择所需的 APB UART 波特率。
 - 分频值 = $UCLK / (\text{所需波特率} \times OSCR)$
 - OSCR 的默认值为 16。
3. 重新设置线路控制寄存器的 DLAB 位为 0。

步骤 2 设置线路控制寄存器。

1. 禁用校验：设置 $PEN = 0$ ；
2. 设置停止位为 1：设置 $STB = 0$ ；
3. 设置字长为 8 位：设置 $WLS = 3$ 。

7.2 发送数据

通过 APB UART 发送数据的步骤如下所示，包括不使用 DMA 和使用 DMA 两种方式。

7.2.1 数据传输（不使用 DMA）

1. 设置 FIFO 控制寄存器的 FIFOE 位，启用 FIFO。

2. 等待 FIFO 清空，轮询线路状态寄存器的 THRE 位，直到其变为 1。
3. 向发送保持寄存器写入最多 FIFO_DEPTH 字节的数据。
4. 如果还有更多的数据需要发送，返回到步骤 2。

7.2.2 数据传输（使用 DMA）

1. 按照以下的值来配置 FIFO 控制寄存器：
 - 指定阈值：设置 TFIFOT；
 - 启用 DMA：设置 DMAE = 1；
 - 启用 FIFO：设置 FIFOE = 1。
2. 设置 DMA 控制器，向发送保持寄存器写入数据：
 - 设置发送保持寄存器为 DMA 的目标地址；
 - 设置突发传输大小为 1；
 - 设置目标数据宽度为 1 字节；
 - 启用硬件握手；
 - 设置 DMA 终端计数中断；
 - 设置其它 DMA 传输控制寄存器。
3. DMA 数据传输在后台执行。当数据输出完成时，DMA 控制器将发出一个终端计数中断。
4. 如果需要确保数据传输完成，请通过检查线路状态寄存器的 THRE 位。

7.3 接收数据

通过 APB UART 接收数据的步骤如下所示，包括不使用 DMA 和使用 DMA 两种方式。

7.3.1 数据接收（不使用 DMA）

1. 设置 FIFO 控制寄存器的 FIFOE 位，启用 FIFO。
2. 等待，直到线路状态寄存器的 DR 位变为 1。
3. 从接收缓冲寄存器读取一个字节的的数据。
4. 返回到步骤 2，继续读取更多的数据。

7.3.2 数据接收（使用 DMA）

1. 按照以下的值来配置 FIFO 控制寄存器：
 - 指定阈值：设置 RFIFOT；
 - 启用 DMA：设置 DMAE = 1；
 - 启用 FIFO：设置 FIFOE = 1。
2. 设置 DMA 控制器，从接收缓冲寄存器读取数据：

- 设置接收缓冲寄存器为 DMA 的源地址；
 - 设置突发传输大小为 1；
 - 设置源数据宽度为 1 字节；
 - 启用硬件握手；
 - 设置 DMA 终端计数中断；
 - 设置其它 DMA 传输控制寄存器。
3. DMA 数据传输在后台执行。当数据输出完成时，DMA 控制器将接收到一个终端计数中断。

8 参考设计

详细信息请参见高云半导体网站 Gowin APB UART IP 相关[参考设计](#):

- 硬件参考设计:

- ...\ref_design\FPGA_RefDesign\DK_START_GW5AST138_V1.0\apb_uart

- 软件参考设计:

- ...\ref_design\MCU_RefDesign\apb_uart

