

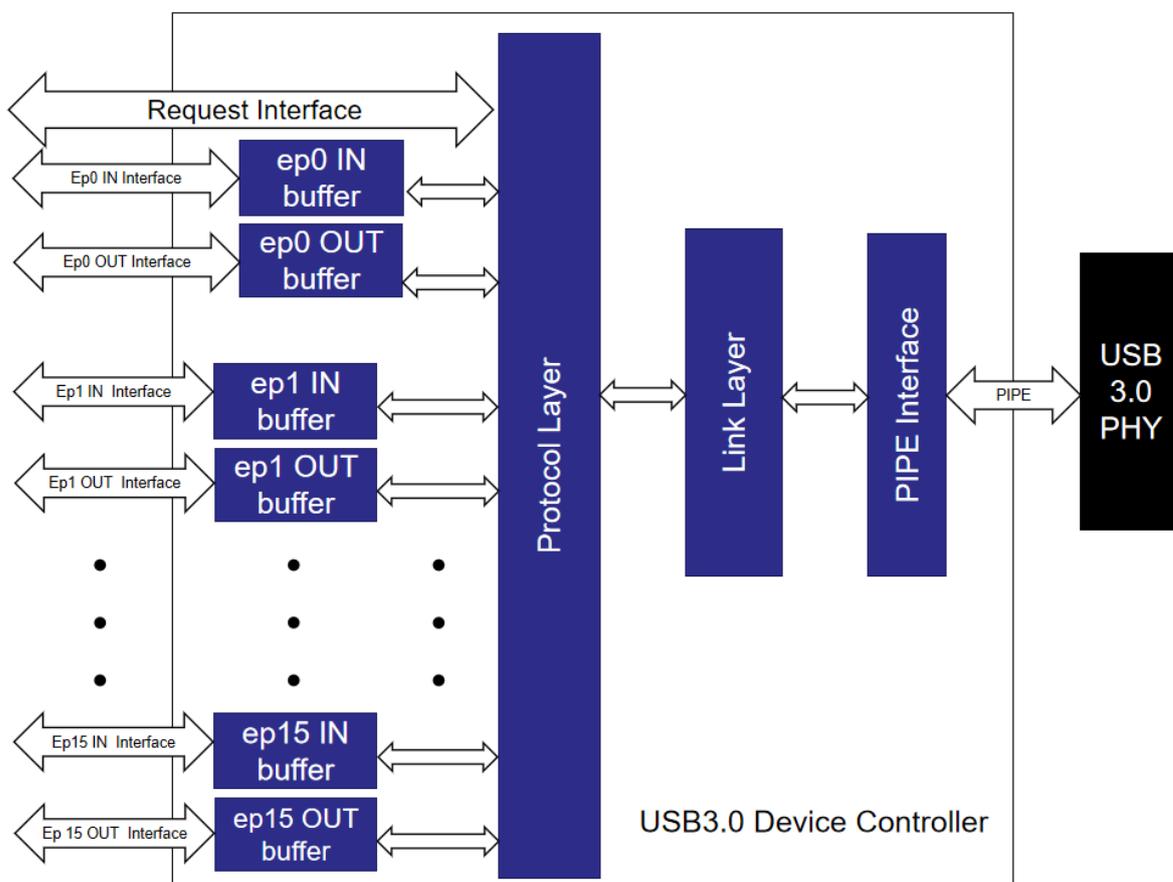
# Gowin USB3.0 Device Controller IP 使用说明

## 功能描述

### USB3.0 Device Controller

USB3.0 Device Controller 位于 User Design 与 PHY 之间。USB3.0 Controller 串联用户设计与 PHY，接收来自 USB 主机端的命令，实现了用户设计及 USB 主机端之间的数据交互。下图为 USB 设备控制器功能框图。

图 1 USB3.0 Device Controller 功能框图



## USB3.0 Device Controller IN Endpoint 用户接口

表 1 Device Controller IN Endpoint 用户接口

接口名称	接口方向	接口位宽	接口功能
epX_in_buf_data_i	I	32	buffer 写数据 位 意义 [31:24] 字节 0 [23:16] 字节 1 [15:8] 字节 2 [7:0] 字节 3
epX_in_buf_wren_i	I	1	buffer 写使能信号 若置 1, 则将在下一个时钟沿将 epX_in_buf_data_i 写入 IN buffer。
epX_in_buf_eob_i	I	1	数据包突发结束信号 数据包写完成信号有效时, Device Controller 将此信号值写入数据包的 eob/lpf 字段。
epX_in_buf_data_commit_i	I	1	数据包写完成信号 每次完整写入一包数据后, 需要将其拉高 1 个时钟周期。Device Controller 会在下一次被轮询时将该包数据发送给主机。
epX_in_buf_data_commit_len_i	I	11	此数据包长度, 范围 0~1024 写完成信号置 1 后, Device Controller 在下一个时钟周期锁存此数据包长度。
epX_in_buf_ready_o	O	1	buffer 空闲信号 若为 1, 表示允许写入一包数据, 否则, 不允许写入。

注!

X 表示具体端点。

Device Controller 内部为 IN 端点分配 IN buffer 用于缓存待发送的数据包, 最多缓存 N 包数据, (N 等于对应端点的最大突发数据包数)。若 epX\_in\_buf\_ready\_o 为 1, 则表示 IN buffer 空闲, 可以存入一包数据。否则不可写入数据包。

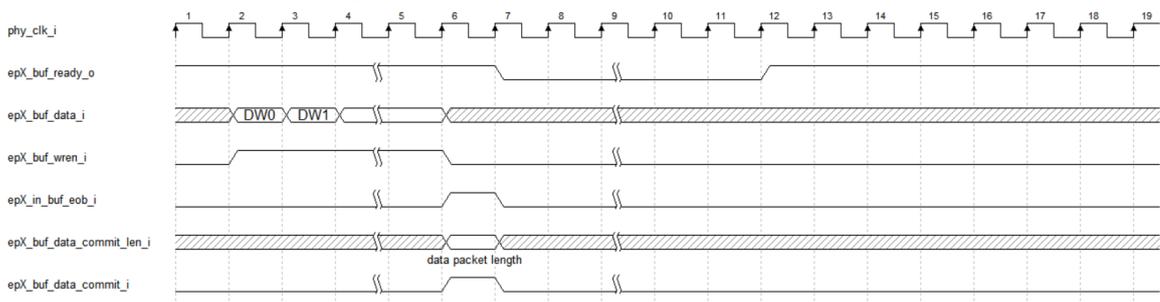
用户通过拉高 epX\_in\_buf\_wren\_i 写入数据。当用户完整写入一包数据后, 需要拉高 epX\_in\_buf\_data\_commit\_i 一个时钟周期, 并将 epX\_in\_buf\_data\_commit\_len\_i 置为该数据包的长度, 并设置好

epX\_in\_buf\_eob\_i。Device Controller 会在下一次该端点被轮询时将该数据包读出并发送给主机。

当主机轮询时，若被轮询端点的 IN buffer 中没有完整数据包，则 Device Controller 会根据端点类型自动回复主机。若为同步端点，Device Controller 不作回复，否则回复 NRDY。

下图是 IN Endpoint 用户接口时序图，该图为用户向端点 X 写入一包数据的过程，且该数据包为一次突发传输的最后一包数据。

图 2 USB3.0 IN Endpoint 用户接口时序图



## USB3.0 Device Controller OUT Endpoint 用户接口

表 2 Device Controller OUT Endpoint 用户接口

接口名称	接口方向	接口位宽	接口功能
epX_out_buf_data_o	O	32	buffer 读数据。 位 意义 [31:24] 字节 0 [23:16] 字节 1 [15:8] 字节 2 [7:0] 字节 3
epX_out_buf_rden_i	I	1	buffer 读使能 拉高后，Device Controller 在下一个时钟周期将数据置于 epX_buf_data_o。
epX_out_buf_len_o	O	11	此数据包长度
epX_out_buf_has_data_o	O	1	buffer 有数据包信号 为 1 时，表示 buffer 中存在一包数据，该包数据长度由 epX_buf_len_o 指示。否则，表示 buffer 中没有数据包。
epX_out_buf_data_ack_i	I	1	数据包读取完成信号 用户将一包数据读取完成后，将此信号拉高一个时钟周期。

**注!**

X 表示具体端点。

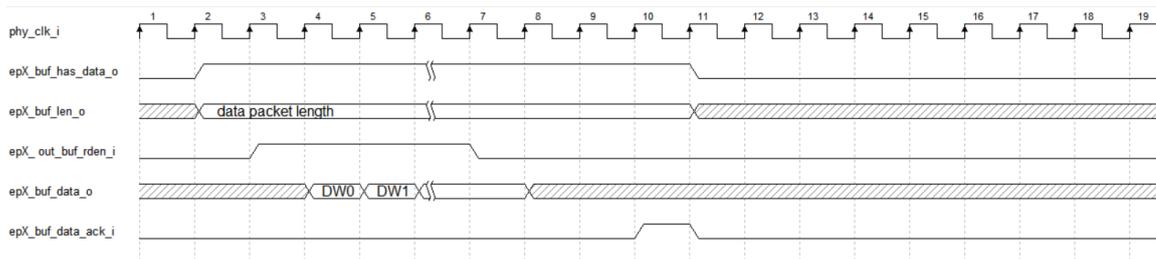
Device Controller 内部为 OUT 端点分配 OUT buffer 用于缓存收到的数据包。OUT buffer 最多缓存 N 个数据包 (N 等于对应端点的最大突发数据包数)。当 Device Controller 收到主机下发的数据包后, 会将 epX\_out\_buf\_has\_data\_o 置 1, 并将 epX\_out\_buf\_len\_o 置为该数据包长度。

当 epX\_out\_buf\_has\_data\_o 为 1 时, 用户可以拉高 epX\_out\_buf\_rden\_i 读取数据包。当用户完成一包数据的读取后, 需要将拉高 epX\_out\_buf\_data\_ack\_i 一个时钟周期。

当主机发来数据包时, 若被轮询端点的 OUT buffer 中没有空闲空间, 则 Device Controller 会根据端点类型自动回复主机。若为同步端点, Device Controller 不作回复, 否则回复 NRDY。

下图是 OUT Endpoint 用户接口时序图, 该图为用户从端点 X 读取出一包数据的过程。

**图 3 USB3.0 OUT Endpoint 用户接口时序图**



## USB3.0 Device Controller Request 用户接口

此用户接口专用于控制传输, 即端点 0 的数据传输。

### Request 用户接口

**表 3 Ep0 SETUP 用户接口**

接口名称	接口方向	接口位宽	接口功能
request_active_o	O	1	接收请求有效信号, 高电平有效; 有效时, 表示设备收到主机发来的请求。
bmRequestType_o	O	8	request_active_o 有效时, 表示所收到 Setup 包的 bmRequestType 字段, 该值会保持不变直至接收到新的请求。
bRequest_o	O	8	request_active_o 有效时, 表示所收到 Setup 包的 bRequest 字段, 该值会保持不变直至接收到新的请求。
wValue_o	O	16	request_active_o 有效时, 表示所收到 Setup 包的

接口名称	接口方向	接口位宽	接口功能
			wValue 字段，该值会保持不变直至接收到新的请求。
wIndex_o	O	16	request_active_o 有效时，表示所收到 Setup 包的 wIndex 字段，该值会保持不变直至接收到新的请求。
wLength_o	O	16	request_active_o 有效时，表示所收到 Setup 包的 wLength 字段，该值会保持不变直至接收到新的请求。

USB3.0 控制传输包含三个阶段，分别是设置阶段，数据阶段以及状态阶段，其中数据阶段可能不存在。

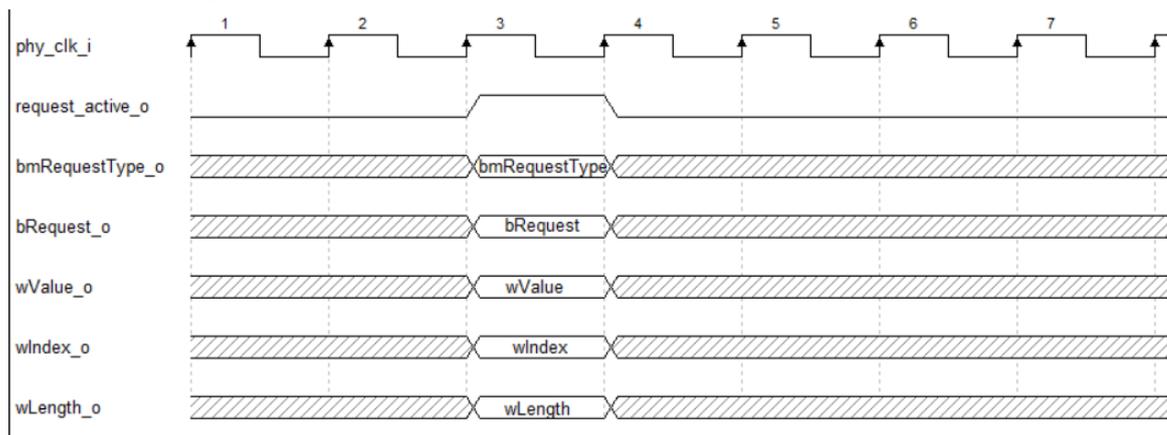
在设置阶段，Device Controller 接收到 Host 发送的 Setup 包后，会自动进行解析。若收到的请求为 SET ADDRESS，Device Controller 会接收被分配的地址，并通过该地址与主机进行通讯。若收到的请求不是 SET ADDRESS，该请求会通过此接口传递给用户。若该请求存在数据阶段，则 Device Controller 自动进入数据阶段，否则进入握手阶段。

在数据阶段，若主机需要发送数据到设备，则 Device Controller 会将接收到的数据通过 Ep0 OUT 用户接口传递给用户。若主机需要从设备读取数据，则用户需要将数据通过 Ep0 IN 用户接口写入，Device Controller 会读取并发送给主机。

在握手阶段，Device Controller 会自动处理，用户无需处理此阶段。

下图是在设置阶段 Request 用户接口时序图,该图表示 Device Controller 向用户传递 Set up 包的过程。

图 4 Request 用户接口时序图



### Ep0 IN 用户接口

参考 USB3.0 Device Controller IN Endpoint 用户接口。

### Ep0 OUT 用户接口

参考 USB3.0 Device Controller OUT Endpoint 用户接口。

## USB3.0 Device Controller PIPE 接口

表 4 Device Controller PIPE 接口

接口名称	接口方向	接口位宽	接口功能
phy_clk_i	I	1	连接 PIPE 接口 PCLK
phy_pipe_rx_data_i	I	32	连接 PIPE 接口 RxData[31:0]
phy_pipe_rx_datak_i	I	4	连接 PIPE 接口 RxDataK[3:0]
phy_pipe_rx_valid_i	I	1	连接 PIPE 接口 RxValid
phy_pipe_tx_data_o	O	32	连接 PIPE 接口 TxData[31:0]
phy_pipe_tx_datak_o	O	4	连接 PIPE 接口 TxDataK[3:0]
phy_reset_n_o	O	1	连接 PIPE 接口 Reset#
phy_tx_detrx_lpbk_o	O	1	连接 PIPE 接口 TxDetectRx/Loopback
phy_tx_elecidle_o	O	1	连接 PIPE 接口 TxElecidle
phy_rx_elecidle_i	I	1	连接 PIPE 接口 RxElecidle
phy_rx_status_i	I	3	连接 PIPE 接口 RxStatus[2:0]
phy_power_down_o	O	2	连接 PIPE 接口 PowerDown[1:0]
phy_phy_status_i	I	1	连接 PIPE 接口 PhyStatus
phy_pwrpresent_i	I	1	连接 PIPE 接口 PowerPresent
phy_tx_oneszeros_o	I	1	连接 PIPE 接口 TxOnesZeros
phy_tx_deemph_o	O	2	连接 PIPE 接口 TxDeemph[1:0]
phy_tx_margin_o	O	3	连接 PIPE 接口 TxMargin[2:0]
phy_tx_swing_o	O	1	连接 PIPE 接口 TxSwing
phy_rx_polarity_o	O	1	连接 PIPE 接口 RxPolarity
phy_rx_termination_o	O	1	连接 PIPE 接口 RX Termination
phy_rate_o	O	1	连接 PIPE 接口 Rate
phy_elas_buf_mode_o	O	1	连接 PIPE 接口 Elasticity Buffer Mode

此接口可以与任意支持 PIPE 接口的 USB3.0 PHY 对接，USB3.0 PHY 中 PCLK 应为 125MHZ，数据总线位宽应为 32bit 模式。

## 信号定义

Gowin USB3.0 Device Controller IP 信号定义如下表所示。

表 5 信号定义

序号	信号名称	方向	位宽	描述
1	phy_clk_i	I	1	连接 PIPE 接口 PCLK
2	phy_pipe_rx_data_i	I	32	连接 PIPE 接口 RxData[31:0]
3	phy_pipe_rx_datak_i	I	4	连接 PIPE 接口 RxDataK[3:0]
4	phy_pipe_rx_valid_i	I	1	连接 PIPE 接口 RxValid
5	phy_pipe_tx_data_o	O	32	连接 PIPE 接口 TxData[31:0]
6	phy_pipe_tx_datak_o	O	4	连接 PIPE 接口 TxDataK[3:0]
7	phy_reset_n_o	O	1	连接 PIPE 接口 Reset#
8	phy_tx_detrx_lpbk_o	O	1	连接 PIPE 接口 TxDetectRx/ Loopback
9	phy_tx_elecidle_o	O	1	连接 PIPE 接口 TxElecidle
10	phy_rx_elecidle_i	I	1	连接 PIPE 接口 RxElecidle
11	phy_rx_status_i	I	3	连接 PIPE 接口 RxStatus[2:0]
12	phy_power_down_o	O	2	连接 PIPE 接口 PowerDown[1:0]
13	phy_phy_status_i	I	1	连接 PIPE 接口 PhyStatus
14	phy_pwrpresent_i	I	1	连接 PIPE 接口 PowerPresent
15	phy_tx_oneszeros_o	I	1	连接 PIPE 接口 TxOnesZeros
16	phy_tx_deemph_o	O	2	连接 PIPE 接口 TxDeemph[1:0]
17	phy_tx_margin_o	O	3	连接 PIPE 接口 TxMargin[2:0]
18	phy_tx_swing_o	O	1	连接 PIPE 接口 TxSwing
19	phy_rx_polarity_o	O	1	连接 PIPE 接口 RxPolarity
20	phy_rx_termination_o	O	1	连接 PIPE 接口 RX Termination
21	phy_rate_o	O	1	连接 PIPE 接口 Rate
22	phy_elas_buf_mode_o	O	1	连接 PIPE 接口 Elasticity Buffer Mode
23	request_active_o	O	1	接收请求有效信号，高电平有效。 有效时，表示设备收到主机发来的请求。
24	bmRequestType_o	O	8	request_active_o 有效时， bmRequestType_o 的值等于所收到 Setup 包的 bmRequestType 字段值。
25	bRequest_o	O	8	request_active_o 有效时，bRequest_o 的 值等于所收到 Setup 包的 bRequest 字段值。

序号	信号名称	方向	位宽	描述
26	wValue_o	O	16	request_active_o 有效时, wValue_o 的值等于所收到 Setup 包的 wValue 字段值。
27	wIndex_o	O	16	request_active_o 有效时, wIndex_o 的值等于所收到 Setup 包的 wIndex 字段值。
28	wLength_o	O	16	request_active_o 有效时, wLength_o 的值等于所收到 Setup 包的 wLength 字段值。
29	epX_in_buf_data_i	I	32	buffer 写数据。 位 意义 [31:24] 字节 0 [23:16] 字节 1 [15:8] 字节 2 [7:0] 字节 3
30	epX_in_buf_wren_i	I	1	buffer 写使能信号 若置 1, 则将在下一个时钟沿将 epX_in_buf_data_i 写入 IN buffer。
31	epX_in_buf_eob_i	I	1	数据包突发结束信号 数据包写完成信号有效时, Device Controller 将此信号值写入数据包的 eob/lpf 字段。
32	epX_in_buf_data_commit_i	I	1	数据包写完成信号 每次完整写入一包数据后, 需要将其拉高 1 个时钟周期。Device Controller 会在下一次被轮询时将该包数据发送给主机。
33	epX_in_buf_data_commit_length_i	I	11	此数据包长度, 范围 0~1024; 写完成信号置 1 后, Device Controller 在下一个时钟周期锁存此数据包长度。
34	epX_in_buf_ready_o	O	1	buffer 空闲信号 若为 1, 表示允许写入一包数据, 否则, 不允许写入。
35	epX_out_buf_data_o	O	32	buffer 读数据。 位 意义 [31:24] 数据包第 i 字节 [23:16] 数据包第 i+1 字节 [15:8] 数据包第 i+2 字节 [7:0] 数据包第 i+3 字节

序号	信号名称	方向	位宽	描述
36	epX_out_buf_rden_i	I	1	buffer 读使能。 拉高后，Device Controller 在下一个时钟周期将数据置于 epX_buf_data_o。
37	epX_out_buf_len_o	O	11	此数据包长度
38	epX_out_buf_has_data_o	O	1	buffer 有数据包信号 为 1 时，表示 buffer 中存在一包数据，该包数据长度由 epX_buf_len_o 指示。否则，表示 buffer 中没有数据包。
39	epX_out_buf_data_ack_i	I	1	数据包读取完成信号 用户将一包数据读取完成后，将此信号拉高一个时钟周期。
40	warm_or_hot_reset_o	O	1	主机复位信号，高电平有效； 有效时，表示主机发来 warm reset 或 hot reset。
41	host_requests_data_from_endpt_o	O	1	主机请求数据信号，高电平有效； 有效时，表示主机轮询到 IN 端点，若此时 Device Controller 内部 IN buffer 存在一包数据，则 Device Controller 将此数据包回复主机。否则，Device Controller 回复 NEDY。
42	host_requests_endpt_num_o	O	4	主机轮询数据端点地址 当 host_requests_data_from_endpt_o 有效时，此信号表示主机轮询到的具体 IN 端点地址。

**注!**

X 表示具体端点。

## 参数配置选项

文件 `usb3_macro_define.v` 中包含了设备端点号以及端点参数。

**表 6 配置选项说明**

参数	描述										
ENDPTX_IN	若定义该参数，则启用 IN 端点 X。										
ENDPTX_IN_TYPE	若启用 IN 端点 X，则该参数对应 IN 端点 X 的传输方式： <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>意义</th> </tr> </thead> <tbody> <tr> <td>01b</td> <td>批量传输</td> </tr> <tr> <td>10b</td> <td>中断传输</td> </tr> <tr> <td>11b</td> <td>同步传输</td> </tr> <tr> <td>00b</td> <td>无定义</td> </tr> </tbody> </table>	值	意义	01b	批量传输	10b	中断传输	11b	同步传输	00b	无定义
值	意义										
01b	批量传输										
10b	中断传输										
11b	同步传输										
00b	无定义										
ENDPTX_IN_BURST	若启用 IN 端点 X，则该参数对应 IN 端点 X 的最大突发个数，范围为 1~16。										
ENDPTX_IN_MAX	若启用 IN 端点 X，则该参数对应 IN 端点 X 的最大数据包长度，范围为 1~1024。										
ENDPTX_OUT	若定义该参数，则启用 OUT 端点 X。										
ENDPTX_OUT_TYPE	若启用 OUT 端点 X，则该参数对应 OUT 端点 X 的传输方式： <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>意义</th> </tr> </thead> <tbody> <tr> <td>01b</td> <td>批量传输</td> </tr> <tr> <td>10b</td> <td>中断传输</td> </tr> <tr> <td>11b</td> <td>同步传输</td> </tr> <tr> <td>00b</td> <td>无定义</td> </tr> </tbody> </table>	值	意义	01b	批量传输	10b	中断传输	11b	同步传输	00b	无定义
值	意义										
01b	批量传输										
10b	中断传输										
11b	同步传输										
00b	无定义										
ENDPTX_OUT_BURST	若启用 OUT 端点 X，则该参数对应 OUT 端点 X 的最大突发个数，范围为 1~16。										
ENDPTX_OUT_MAX	若启用 OUT 端点 X，则该参数对应 OUT 端点 X 的最大数据包长度，范围为 1~1024。										

**注!**

ENDPTX 中的 X 代表端点号，范围为 1~15。

## 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](http://www.gowinsemi.com)

E-mail：[support@gowinsemi.com](mailto:support@gowinsemi.com)

Tel: +86 755 8262 0391

## 版本信息

日期	版本	说明
08/13/2024	1.0	初始版本。
01/17/2025	1.1	更新表 6 配置选项说明。

版权所有 © 2025 广东高云半导体科技股份有限公司

**GOWIN高云**、Gowin 以及高云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。