



Gowin PicoRV32 软件编程参考手册

IPUG911-1.1,2020-03-12

版权所有© 2020 广东高云半导体科技股份有限公司

未经本公司书面许可，任何单位和个人不得擅自摘抄、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，或以禁止发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档进行适时的更新。

版本信息

日期	版本	说明
2020/01/16	1.0	初始版本。
2020/03/12	1.1	<ul style="list-style-type: none">● MCU 支持 Wishbone 总线接口的外部设备 GPIO;● MCU 支持扩展 AHB 总线接口;● MCU 支持片外 SPI-Flash 下载及运行;● MCU 支持外部设备 SPI-Flash 读、写和擦除功能;● MCU 支持 Hardware Stack Protection 和 Trap Stack Overflow 功能。

目录

目录	i
表目录	iii
1 软件编程库	1
2 存储系统	2
3 中断处理	3
3.1 中断特征	3
3.2 中断控制	3
3.3 中断编号	3
4 Simple UART	5
4.1 特征	5
4.2 寄存器定义	5
4.3 驱动程序使用方法	5
5 Wishbone I2C Master	6
5.1 特征	6
5.2 寄存器定义	6
5.3 驱动程序使用方法	7
6 Wishbone SPI Master/Slave	8
6.1 特征	8
6.2 寄存器定义	8
6.3 驱动程序使用方法	9
7 Wishbone UART	10
7.1 特征	10
7.2 寄存器定义	10
7.3 驱动程序使用方法	10
8 Wishbone GPIO	11
8.1 特征	11
8.2 寄存器定义	11
8.3 驱动程序使用方法	11

9 ADV SPI-Flash	12
9.1 特征	12
9.2 寄存器定义	12
9.3 驱动程序使用方法	16

表目录

表 1-1 Gowin_PicoRV32 软件编程库	1
表 2-1 Gowin_PicoRV32 标准外设内存映射	2
表 3-1 Gowin_PicoRV32 中断控制函数.....	3
表 3-2 Gowin_PicoRV32 中断编号	3
表 4-1 UART 寄存器定义.....	5
表 4-2 Simple UART 驱动程序使用方法.....	5
表 5-1 Wishbone I2C Master 寄存器定义.....	6
表 5-2 Wishbone I2C Master 驱动程序使用方法.....	7
表 6-1 Wishbone SPI 寄存器定义.....	8
表 6-2 Wishbone SPI 驱动程序使用方法.....	9
表 7-1 Wishbone UART 寄存器定义.....	10
表 7-2 Wishbone UART 驱动程序使用方法.....	10
表 8-1 Wishbone GPIO 寄存器定义	11
表 8-2 Wishbone GPIO 驱动程序使用方法	11
表 9-1 SPI-Flash 寄存器定义.....	12
表 9-2 SPI-Flash 驱动程序使用方法.....	16

1 软件编程库

Gowin_PicoRV32 提供软件编程库 Gowin_PicoRV32\src\c_lib, 如表 1-1 所示。

表 1-1 Gowin_PicoRV32 软件编程库

文件	描述
start.S	Gowin_PicoRV32启动引导程序
custom_ops.S	自定义中断处理指令宏定义
config.h	用户配置启动运行方式
picorv32.h	寄存器、地址映射、时钟信号、系统配置等定义
irq.c	中断处理函数
sections.lds	Flash链接器： MCU boot from external Flash and run in ITCM和MCU boot and run in ITCM方式
sections_xip.lds	Flash链接器： MCU boot and run in external Flash方式
firmware.c	性能统计、延时等定义
loader.c	启动引导方式定义
printf.c	Simple UART/WB UART打印输出
simpleuart.c	Simple UART的驱动函数定义
wbi2c.c	Wishbone I2C Master的驱动函数定义
wbspi.c	Wishbone SPI Master/Slave的驱动函数定义
wbuart.c	Wishbone UART的驱动函数定义
wbgpio.c	Wishbone GPIO的驱动函数定义
advspi.c	ADV SPI-Flash的驱动函数定义

2 存储系统

Gowin_PicoRV32 标准外部设备内存映射地址，如表 2-1 所示。

Gowin_PicoRV32 标准外部设备内存映射地址定义位置：picorv32.h。

表 2-1 Gowin_PicoRV32 标准外设内存映射

标准外设	类型	地址映射	描述
DTCM	-	0x00000000	8KB, 16KB, 32KB, 64KB, 128KB, 256KB
ITCM	-	0x01000000	8KB, 16KB, 32KB, 64KB, 128KB, 256KB
SIMPLEUART	SIMPLEUART_RegDef	0x02000000	Simple UART
WBUART	WBUART_RegDef	0x11000000	Wishbone UART
WBGPIO	WBGPIO_RegDef	0x11001000	Wishbone GPIO
WBSPI_MASTER	WBSPI_RegDef	0x11002000	Wishbone SPI Master
WBSPI_SLAVE	WBSPI_RegDef	0x11003000	Wishbone SPI Slave
ADVSPi_FLASH	SPI_FLASH_RegDef	0x11004000	ADV SPI-Flash
WBI2C_Master	WBI2CMaster_RegDef	0x11005000	Wishbone I2C Master
OPEN_WB_INTERFACE	-	0x20000000	Open Wishbone Interface for Customized Peripherals
OPEN_AHB_INTERFACE	-	0x80000000	Open AHB Interface for Customized Peripherals

3 中断处理

3.1 中断特征

中断控制器包括以下特征：

- 轻量化的中断控制器和中断控制指令
- 中断由上升沿触发
- 提供最多 32 个用户可用的中断处理信号
- 软件可控制的中断优先级
- 不支持中断嵌套

3.2 中断控制

Gowin_PicoRV32 中断控制函数，如表 3-1 所示。

表 3-1 Gowin_PicoRV32 中断控制函数

中断控制函数	参数	描述
mask_irq	32位屏蔽码	中断屏蔽函数 屏蔽码中为0的位对应的中断为开放状态； 屏蔽码中为1的位对应的中断为屏蔽状。
irq_mask_one_bit	32位中断编号	单个中断屏蔽函数 该编号对应的中断位被屏蔽
irq_unmask_one_bit	32位中断编号	单个中断解除屏蔽函数 该编号对应的中断位解除屏蔽
set_timer	32位的计时长度 timer_val	计时器中断设置函数 自动开始计时，当计时达到timer_val个时钟周期时，触发 计时器中断。
irq	32位中断寄存器 32位中断编号	中断处理函数： 当触发中断后，自动进入该函数，根据触发中断的中断位 编号，调用相应的中断处理函数。

3.3 中断编号

Gowin_PicoRV32 预置的中断编号，如表 3-2 所示。

Gowin_PicoRV32 中断处理函数定义位置：irq.c。

表 3-2 Gowin_PicoRV32 中断编号

编号	描述
0	32位定时器

编号	描述
1	EBREAK/ECALL指令中断
2	内存访问非对齐错误
3	硬件栈保护
4	Wishbone SPI Master中断
5	Wishbone SPI Slave中断
6	Wishbone I2C Master中断
20 ~ 31	预留给Wishbone扩展总线接口和AHB扩展总线接口，供用户扩展Wishbone和AHB外部设备使用

4 Simple UART

4.1 特征

Gowin_PicoRV32 包含一个轻量级的通用异步收发器 Simple UART:

- 无奇偶校验位
- 8 位数据位
- 1 位停止位
- 不支持中断

用户在使能 Simple UART 时, 需设置波特率分频寄存器, 例如, 如果系统时钟频率运行在 12MHz, 需要波特率为 9600, 则可以设置波特率分频寄存器为 $12000000/9600=1250$ 。

4.2 寄存器定义

Simple UART 寄存器定义如表 4-1 所示。

表 4-1 UART 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CLKDIV	0x004	RW	32	0x1	分频系数寄存器, 用于配置波特率
DATA	0x008	RW	32	0x-	输入/输出数据寄存器

4.3 驱动程序使用方法

Simple UART 驱动程序使用方法如表 4-2 所示。

表 4-2 Simple UART 驱动程序使用方法

名称	描述
uart_init	对 Simple UART 进行初始化, 配置波特率
outbyte	输出一个字符, 当输出换行符时自动返回行首。
uart_putchar	输出一个字符
getchar_prompt	返回 UART RX 接收到的字符
uart_getchar	返回 UART RX 接收到的字符

5 Wishbone I2C Master

5.1 特征

Gowin_PicoRV32 包含一个通过 Wishbone 总线访问的内部集成电路 I2C Master 模块：

- Wishbone 总线接口
- 符合业界标准的 I2C 总线协议
- 总线仲裁及仲裁丢失检测
- 总线忙状态检测
- 产生中断标志
- 产生起始、终止、重复起始和应答信息
- 支持起始、终止和重复起始检测
- 支持 7 位寻址模式

5.2 寄存器定义

Wishbone I2C Master 寄存器定义如表 5-1 所示。

表 5-1 Wishbone I2C Master 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
PRER	0x00	RW	32	0x0000 FFFF	Clock prescale register [31:15] Reserved [15:0] Prescale value = sys_clk/(5*SCL)-1
CTR	0x04	RW	32	0x0000 0000	Control register [31:8] Reserved [7] Enable I2C function [6] Enable I2C interrupt [5:0] Reserved
TXR	0x08	WO	32	0x0000 0000	Transmit data register [31:8] Reserved [7:1] Next transmission data [0] Data direction
RXR	0x08	RO	32	0x0000 0000	Receive data register [31:8] Reserved [7:0] Last received data
CR	0x0c	WO	32	0x0000 0000	Command register [31:8] Reserved [7] STA, Start transmission status

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[6] STO, Over transmission status [5] RD, Read enable, read data from slave [4] WR, Write enable, write data to slave [3] Acknowledge [2:1] Reserved [0] Interrupt acknowledge
SR	0x0c	RO	32	0x0000 0000	Status register [31:8] Reserved [7] Receive acknowledge signal from slave [6] I2C busy status [5] Arbitration loss [4:2] Reserved [1] Data transmission status flag [0] Interrupt flag

5.3 驱动程序使用方法

Wishbone I2C Master 驱动程序使用方法如表 5-2 所示。

表 5-2 Wishbone I2C Master 驱动程序使用方法

名称	描述
I2C_Init	I2C Initialization
I2C_SendByte	Send a byte to I2C bus
I2C_SendBytes	Send multiple bytes to I2C bus
I2C_SendWord	Send a word to I2C bus
I2C_SendArray	Send an array bytes to I2C bus
I2C_ReceiveByte	Read a byte from I2C bus
I2C_ReadBytes	Read multiple bytes from I2C bus
I2C_ReceiveWord	Read a word from I2C bus
I2C_RecevieArray	Read an array bytes from I2C bus
I2C_Rate_Set	Set I2C traffic rate
I2C_Enable	Enable I2C bus
I2C_Disable	Disable I2C bus
I2C_InterruptOpen	Open I2C interrupt
I2C_InterruptClose	Close I2C interrupt

6 Wishbone SPI Master/Slave

6.1 特征

Gowin_PicoRV32 包含一个通过 Wishbone 总线访问的串行外设 Master 接口 SPI Master 和一个通过 Wishbone 总线访问的串行外设 Slave 接口 SPI Slave:

- Wishbone 总线接口
- 全双工同步串行数据传输
- 可配置时钟极性和相位
- SPI 产生的串行时钟频率可配置
- 可配置的数据接收寄存器和数据发送寄存器位宽

6.2 寄存器定义

Wishbone SPI 寄存器定义如表 6-1 所示。

表 6-1 Wishbone SPI 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RXDATA	0x00	RO	32	0x0000 0000	Receive data register [31:8] Reserved [7:0] Receive data
TXDATA	0x04	WO	32	0x0000 0000	Transmit data register [31:8] Reserved [7:0] Transmit data
STATUS	0x08	RW	32	0x0000 0000	Status register [31:8] Reserved [7] Overflow error status [6] Receive ready status [5] Transmit ready status [4] Be transmitting [3] Transmit overrun error status [2] Receive overrun error status [1:0] Reserved
CONTROL	0x0C	RW	32	0x0000 0000	Control register [31:5] Reserved [4:3] Clock selected, CLK_I / 2/4/6/8 [2] Clock polarity [1] Clock phase [0] Direction, 1 is MSB first

寄存器名称	地址偏移	类型	宽度	初始值	描述
SSMASK	0x10	RW	32	0x0000 0000	[31:1] Reserved [0] Select and enable slave

6.3 驱动程序使用方法

Wishbone SPI 驱动程序使用方法如表 6-2 所示。

表 6-2 Wishbone SPI 驱动程序使用方法

名称	描述
wbspi_master_select_slave	WBSPI Master选择一个从设备进行通信
wbspi_enable_interrupt	开启WBSPI的中断
wbspi_disable_interrupt	关闭WBSPI的中断
wbspi_master_txdata	WBSPI Master向从设备发送数据
wbspi_master_rxdata	WBSPI Master读取从设备的数据
wbspi_slave_prepare_txdata	WBSPI Slave预备数据，等待主设备读取
wbspi_slave_read_data	WBSPI Slave读取主设备发送过来的数据

7 Wishbone UART

7.1 特征

Gowin_PicoRV32 包含一个通过 Wishbone 总线访问的 UART:

- Wishbone 总线接口
- 无奇偶校验位
- 8 位数据位
- 1 位停止位

7.2 寄存器定义

Wishbone UART 寄存器定义如表 7-1 所示。

表 7-1 Wishbone UART 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
SETUP	0x00	RW	32	0x00000000	UART参数/设置寄存器
FIFO	0x04	RO	32	0x00000000	输入FIFO和输出FIFO的状态寄存器
RXREG	0x08	RO	32	0x00000000	UART接受数据寄存器
TXREG	0x0C	RW	32	0x00000000	UART发送数据寄存器

7.3 驱动程序使用方法

Wishbone UART 驱动程序使用方法如表 7-2 所示。

表 7-2 Wishbone UART 驱动程序使用方法

名称	描述
wbuart_init	Wishbone UART初始化, 配置波特率
wbuart_putc	Wishbone UART发送一个字节
wbuart_getc	Wishbone UART接受一个字节
wbuart_outbyte	Wishbone UART发送一个字节, 并在输出换行时自动返回行首

8 Wishbone GPIO

8.1 特征

Gowin_PicoRV32 包含一个通过 Wishbone 总线访问的 GPIO:

- Wishbone 总线接口
- 32 位，每位可以独立配置输入和输出状态

8.2 寄存器定义

Wishbone GPIO 寄存器定义如表 8-1 所示。

表 8-1 Wishbone GPIO 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DIR	0x10	RW	32	0xFFFFFFFF	GPIO input/output direction register [31:0] Control each pin input/output direction 1 = Output 0 = Input
IN	0x14	RO	32	0x00000000	GPIO input register [31:0] Each pin input
OUT	0x18	WO	32	0x00000000	GPIO output register [31:0] Each pin output

8.3 驱动程序使用方法

Wishbone GPIO 驱动程序使用方法如表 8-2 所示。

表 8-2 Wishbone GPIO 驱动程序使用方法

名称	描述
GPIO_Init	Wishbone GPIO初始化
GPIO_SetDir	Wishbone GPIO设置输入/输出方向
GPIO_GetDir	Wishbone GPIO获取输入/输出方向
GPIO_EnableWriteBit	Wishbone GPIO使能每位输出
GPIO_EnableReadBit	Wishbone GPIO使能每位输入
GPIO_WriteData	Wishbone GPIO输出
GPIO_ReadData	Wishbone GPIO输入

9 ADV SPI-Flash

9.1 特征

Gowin_PicoRV32 包含 1 个通过 Wishbone 总线访问的 SPI-Flash:

- SPI-Flash 支持软件编程设计 BIN 文件下载启动和指令运行功能
- SPI-Flash Memory 支持读、写和擦除功能
- SPI-Flash Controller 默认为高云开发板板载 Winbond W25Q64BV 芯片

9.2 寄存器定义

SPI-Flash 寄存器定义如表 9-1 所示。

表 9-1 SPI-Flash 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
IDREV	0x00	RO	32	0x02002000	ID and revision register [31:8] ID number [7:4] Major revision number [3:0] Minor revision number
RESERVED0[3]	0x04-0x0C	-	-	-	Reserved
TRANSFMT	0x10	RW	32	0x00020780	SPI transfer format register [31:18] Reserved [17:16] Address length in bytes 00 = 1 byte 01 = 2 bytes 10 = 3 bytes 11 = 4 bytes [15:13] Reserved [12:8] Data length [7] Enable data merge mode [6:5] Reserved [4] Bi-directional MOSI in single mode 0 = MOSI is uni-directional signal 1 = MOSI is bi-directional signal [3] Transfer data with the least significant bit first 0 = Most significant bit first 1 = Least significant bit first [2] SPI master/slave mode selection 0 = Master mode

寄存器名称	地址偏移	类型	宽度	初始值	描述
					1 = Slave mode [1] SPI clock polarity 0 = SCLK is LOW in the idle states 1 = SCLK is HIGH in the idle states [0] SPI clock phase 0 = Sampling data at odd SCLK edges 1 = Sampling data at even SCLK edges
DIRECTIO	0x14	RW	32	0x0	SPI direct IO control register [31:25] Reserved [24] Enable direct IO 0 = Disable 1 = Enable [23:22] Reserved [21] Output enable for SPI-Flash hold signal [20] Output enable for SPI-Flash write protect signal [19] Output enable for the SPI MISO signal [18] Output enable for the SPI MOSI signal [17] Output enable for SPI SCLK signal [16] Output enable for SPI CS signal [15:14] Reserved [13] Output value for SPI-Flash hold signal [12] Output value for SPI-Flash write protect signal [11] Output value for SPI MISO signal [10] Output value for SPI MOSI signal [9] Output value for SPI SCLK signal [8] Output value for SPI CS signal [7:6] Reserved [5] Status of SPI-Flash hold signal [4] Status of SPI-Flash write protect signal [3] Status of SPI MISO signal [2] Status of SPI MOSI signal [1] Status of SPI SCLK signal [0] Status of SPI CS signal
RESERVED1[2]	0x18-0x1C	-	-	-	Reserved
TRANSCTRL	0x20	RW	32	0x0	SPI transfer control register [31] Reserved [30] SPI command phase enable 0 = Disable the command phase 1 = Enable the command phase (Master mode only) [29] SPI address phase enable 0 = Disable the address phase 1 = Enable the address phase (Master mode only) [28] SPI address phase format

寄存器名称	地址偏移	类型	宽度	初始值	描述
					0 = Address phase is single mode 1 = The format of the address phase is the same as the DualQuad data phase (Master mode only) [27:24] Transfer mode 0000 = Write and read at the same time 0001 = Write only 0010 = Read only 0011 = Write, Read 0100 = Read, Write 0101 = Write, Dummy, Read 0110 = Read, Dummy, Write 0111 = None data 1000 = Dummy, Write 1001 = Dummy, Read 1010~1111 = Reserved [23:22] SPI data phase format 00 = Single mode 01 = Dual I/O mode 10 = Quad I/O mode 11 = Reserved [21] Append and one-byte special token following the address phase for SPI read transfers [20:12] Transfer count for write data [11] The value of the one-byte special token following the address phase for SPI read transfers 0 = token value is 0x00 1 = token value is 0x69 [10:9] Dummy data count [8:0] Transfer count for read data
CMD	0x24	RW	32	0x0	SPI command register [31:8] Reserved [7:0] SPI command
ADDR	0x28	RW	32	0x0	SPI address register [31:0] SPI address (Master mode only)
DATA	0x2C	RW	32	0x0	SPI data register [31:0] Data to transmit or the received data
CTRL	0x30	RW	32	0x0	SPI controller register [31:21] Reserved [20:16] Transmit FIFO threshold [15:13] Reserved [12:8] Receive FIFO threshold [7:5] Reserved [4] TX DMA enable [3] RX DMA enable [2] Transmit FIFO reset [1] Receive FIFO reset [0] SPI reset
STATUS	0x34	RO	32	0x0	SPI status register [31:24] Reserved [23] Transmit FIFO full flag [22] Transmit FIFO empty flag

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[21] Reserved [20:16] Number of valid entries int the transmit FIFO [15] Receive FIFO full flag [14] Receive FIFO empty flag [13] Reserved [12:8] Number of valid entries in the receive FIFO [7:1] Reserved [0] SPI register programming is in progress
INTREN	0x38	RW	32	0x0	SPI interrupt enable register [31:6] Reserved [5] Enable the slave command interrupt [4] Enable the end of SPI transfer interrupt [3] Enable the SPI transmit FIFO threshold interrupt [2] Enable the SPI receive FIFO threshold interrupt [1] Enable SPI transmit FIFO underrun interrupt (Slave mode only) [0] Enable SPI receive FIFO overrun interrupt (Slave mode only)
INTRST	0x3C	WO	32	0x0	SPI interrupt status register [31:6] Reserved [5] Slave command interrupt (Slave mode only) [4] End of SPI transfer interrupt [3] TX FIFO threshold interrupt [2] RX FIFO threshold interrupt [1] TX FIFO underrun interrupt (Slave mode only) [0] RX FIFO overrun interrupt (Slave mode only)
TIMING	0x40	RW	32	0x0	SPI interface timing register [31:14] Reserved [13:12] The minimum time between the edges of SPI CS and the edges of SCLK [11:8] The minimum time the SPI CS should stay HIGH [7:0] The clock frequency ratio between the clock source and SPI interface SCLK
RESERVED2[3]	0x44-0x4c	-	-	-	Reserved
MEMCTRL	0x50	RW	32	0x0	SPI memory access control register [31:9] Reserved [8] This bit is set when "MEMCTRL" / "TIMING" is written [7:4] Reserved [3:0] Selects the SPI command
RESERVED3[3]	0x54-0x5C	-	-	-	Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
SLVST	0x60	RW	32	0x0	SPI slave status register [31:19] Reserved [18] Data underrun occurs in the last transaction [17] Data overrun occurs in the last transaction [16] SPI is ready for data transaction [15:0] User defined status flags
SLVDATAcnt	0x64	RO	32	0x0	SPI slave data count register [31:25] Reserved [24:16] Slave transmitted data count [15:9] Reserved [8:0] Slave received data count
RESERVED4[5]	0x68-0x78	-	-	-	Reserved
CONFIG	0x7C	RO	32	0x0	Configuration register [31:15] Reserved [14] Support for SPI slave mode [13] Reserved [12] Support for memory-mapped access through AHB bus [11] Support for direct SPI IO [10] Reserved [9] Support for Quad I/O SPI [9] Support for Dual I/O SPI [7:6] Reserved [5:4] Depth of TX FIFO 00 = 2 words 01 = 4 words 10 = 8 words 11 = 16 words [3:2] Reserved [1:0] Depth of RX FIFO 00 = 2 words 01 = 4 words 10 = 8 words 11 = 16 words

9.3 驱动程序使用方法

SPI-Flash 驱动程序使用方法如表 9-2 所示。

表 9-2 SPI-Flash 驱动程序使用方法

名称	描述
spi_flash_init	Initialize SPI-Flash
change_mode_spi_flash	Switch SPI-Flash mode between download and read, write, erase memory
spi_flash_read	Read data from SPI-Flash
spi_flash_write	Write data into SPI-Flash
spi_flash_page_program	Write data into SPI-Flash with pages
spi_flash_sector_erase	Erase SPI-Flash with sector
spi_flash_write_cmd	Write command to SPI-Flash
spi_flash_read_status	Read SPI-Flash status

