Gowin_EMPU(GW1NS-4C)软件编程

参考手册

IPUG931-1.2,2021-06-21

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
2020/04/20	1.0	初始版本。
2021/02/08	1.1	● 支持外部设备 AHB PSRAM Memory Interface； ● 支持外部设备 AHB HyperRAM Memory Interface； ● 支持外部设备 APB SPI_Nor_Flash； ● GPIO 支持多种端口类型配置； ● I^2C 支持多种端口类型配置； ● 升级软件版本以及软件开发工具包。
2021/06/21	1.2	修复已知的 SPI 全双工读写问题。

目录

图目录

表目录

1 软件编程库

Gowin_EMPU(GW1NS-4C)支持软件编程库：Gowin_EMPU\src\c_lib。

Gowin_EMPU(GW1NS-4C)支持两种软件编程方法：

- MCU 内核 Cortex-M3 软件编程
- 嵌入式 RTOS 软件编程

1.1 MCU 内核 Cortex-M3 软件编程

Gowin_EMPU(GW1NS-4C)软件编程库，支持 MCU 内核 Cortex-M3 软件编程方法，如表 1-1 所示。

表 1-1 MCU 内核 Cortex-M3 软件编程

文件	描述
startup_gw1ns4c.s	MCU内核启动引导程序
core_cm3.c	MCU内核Cortex-M3内核寄存器定义
gw1ns4c.h	中断向量表、寄存器和地址映射定义
system_gw1ns4c.c	系统初始化和系统时钟定义
gw1ns4c_flash.ld	GMD IDE Flash链接器
gw1ns4c_gpio.c	GPIO驱动函数定义
gw1ns4c_uart.c	UART驱动函数定义
gw1ns4c_timer.c	Timer驱动函数定义
gw1ns4c_wdog.c	WatchDog驱动函数定义
gw1ns4c_spi.c	SPI Master驱动函数定义
gw1ns4c_i2c.c	I^2C Master驱动函数定义
gw1ns4c_rtc.c	RTC驱动函数定义
gw1ns4c_misc.c	中断优先级和SysTick定义
gw1ns4c_syscon.c	系统控制驱动函数定义

文件	描述
gw1ns4c_it.c	中断处理函数定义
malloc.c	动态内存管理malloc和free函数重定向定义
retarget.c	UART0 printf函数重定向定义
psram.c	内嵌PSRAM Memory Interface驱动函数定义
hyper_ram.c	内嵌HyperRAM Memory Interface驱动函数定义
spi_nor_flash.c	内嵌SPI_Nor_Flash驱动函数定义

1.2 嵌入式 RTOS 软件编程

Gowin_EMPU(GW1NS-4C)，支持以下两种嵌入式操作系统软件编程：

- uC/OS-III
- FreeRTOS

2 存储系统

2.1 标准外设内存映射

Gowin_EMPU(GW1NS-4C)标准外设内存映射地址定义，如表 2-1 所示。

表 2-1 标准外设内存映射定义

标准外设	类型	地址映射	描述
FLASH	-	0x00000000	32KB
SRAM	-	0x20000000	2KB、4KB、8KB或16KB
TIMER0	TIMER_TypeDef	0x40000000	定时器0
TIMER1	TIMER_TypeDef	0x40001000	定时器1
UART0	UART_TypeDef	0x40004000	通用异步收发器0
UART1	UART_TypeDef	0x40005000	通用异步收发器1
RTC	RTC_TypeDef	0x40006000	实时时钟
WatchDog	WDOG_TypeDef	0x40008000	看门狗
GPIO0	GPIO_TypeDef	0x40010000	通用输入输出端口
SYSCON	SYSCON_TypeDef	0x4001F000	系统控制
I2C	I2C_TypeDef	0x40002000	内部集成电路总线
SPI	SPI_TypeDef	0x40002200	串行外设接口
APB2 Master 1	-	0x40002400	APB2 Master [1]
APB2 Master 2	-	0x40002500	APB2 Master [2]
APB2 Master 3	-	0x40002600	APB2 Master [3]
APB2 Master 4	-	0x40002700	APB2 Master [4]
APB2 Master 5	-	0x40002800	APB2 Master [5]
APB2 Master 6	-	0x40002900	APB2 Master [6]
APB2 Master 7	-	0x40002A00	APB2 Master [7]

标准外设	类型	地址映射	描述
APB2 Master 8	-	0x40002B00	APB2 Master [8]
APB2 Master 9	-	0x40002C00	APB2 Master [9]
APB2 Master 10	-	0x40002D00	APB2 Master [10]
APB2 Master 11	-	0x40002E00	APB2 Master [11]
APB2 Master 12	-	0x40002F00	APB2 Master [12]
AHB2 Master	-	0xA0000000	AHB2 Master

2.2 内核系统内存映射

Gowin_EMPU(GW1NS-4C)内核系统内存映射定义，如表 2-2 所示。

表 2-2 内核系统内存映射定义

系统控制	类型	地址映射	描述
ITM	ITM_Type	0xE0000000	ITM configuration structure
DWT	DWT_Type	0xE0001000	DWT configuration structure
CoreDebug	CoreDebug_Type	0xE000EDF0	Core Debug configuration structure
ETM	ETM_Type	0xE0041000	ETM configuration structure
SysTick	SysTick_Type	0xE000E010	SysTick configuration structure
NVIC	NVIC_BASE	0xE000E100	NVIC configuration structure
SCnSCB	SCnSCB_Type	0xE000E000	System control Register not in SCB
SCB	SCB_Type	0xE000ED00	SCB configuration structure
TPIU	TPIU_Type	0xE0040000	TPIU configuration structure

3 中断处理

嵌套向量中断控制器包括以下特征：

- 支持多达 32 个低延时中断
- 提供 6 个用户可用的中断处理信号（USER_INT0~5）
- 支持 0-7 级可编程中断优先级
- 低延时中断和异常处理
- 中断信号边沿或脉冲检测
- 中断优先级动态调整

Gowin_EMPU(GW1NS-4C)中断控制器定义，如表 3-1 所示。

表 3-1 中断控制器定义

Address	Interrupt	Number	Description
0x00000000	__StackTop	–	Top of Stack
0x00000004	Reset_Handler	–	Reset Handler
0x00000008	NMI_Handler	-14	NMI Handler
0x0000000C	HardFault_Handler	-13	Hard Fault Handler
0x00000010	MemManage_Handler	-12	MPU Fault Handler
0x00000014	BusFault_Handler	-11	Bus Fault Handler
0x00000018	UsageFault_Handler	-10	Usage Fault Handler
0x0000001C	0	–	Reserved
0x00000020	0	–	Reserved
0x00000024	0	–	Reserved
0x00000028	0	–	Reserved
0x0000002C	SVC_Handler	-5	SVCall Handler
0x00000030	DebugMon_Handler	-4	Debug Monitor Handler
0x00000034	0	–	Reserved

Address	Interrupt	Number	Description
0x00000038	PendSV_Handler	-2	PendSV Handler
0x0000003C	SysTick_Handler	-1	SysTick Handler
0x00000040	UART0_Handler	0	16+ 0: UART 0 RX and TX Handler
0x00000044	USER_INT0_Handler	1	16+ 1: USER_INT0
0x00000048	UART1_Handler	2	16+ 2: UART 1 RX and TX Handler
0x0000004C	USER_INT1_Handler	3	16+ 3: USER_INT1
0x00000050	USER_INT2_Handler	4	16+ 4: USER_INT2
0x00000054	RTC_Handler	5	16+ 5: RTC Handler
0x00000058	PORT0_COMB_Handler	6	16+ 6: GPIO Port 0 Combined Handler
0x0000005C	USER_INT3_Handler	7	16+ 7: USER_INT3
0x00000060	TIMER0_Handler	8	16+ 8: TIMER 0 Handler
0x00000064	TIMER1_Handler	9	16+ 9: TIMER 1 Handler
0x00000068	0	–	16+10: Reserved
0x0000006C	I2C_Handler	11	16+11: I2C Handler
0x00000070	UARTOVF_Handler	12	16+12: UART 0,1 Overflow Handler
0x00000074	USER_INT4_Handler	13	16+13: USER_INT4
0x00000078	USER_INT5_Handler	14	16+14: USER_INT5
0x0000007C	Spare15_Handler	15	16+ 15: Not Used
0x00000080	PORT0_0_Handler	16	16+16: GPIO Port 0 pin 0 Handler
0x00000084	PORT0_1_Handler	17	16+17: GPIO Port 0 pin 1 Handler
0x00000088	PORT0_2_Handler	18	16+18: GPIO Port 0 pin 2 Handler
0x0000008C	PORT0_3_Handler	19	16+19: GPIO Port 0 pin 3 Handler
0x00000090	PORT0_4_Handler	20	16+20: GPIO Port 0 pin 4 Handler
0x00000094	PORT0_5_Handler	21	16+21: GPIO Port 0 pin 5 Handler
0x00000098	PORT0_6_Handler	22	16+22: GPIO Port 0 pin 6 Handler
0x0000009C	PORT0_7_Handler	23	16+23: GPIO Port 0 pin 7 Handler
0x000000A0	PORT0_8_Handler	24	16+24: GPIO Port 0 pin 8 Handler
0x000000A4	PORT0_9_Handler	25	16+25: GPIO Port 0 pin 9 Handler
0x000000A8	PORT0_10_Handler	26	16+26: GPIO Port 0 pin 10 Handler
0x000000AC	PORT0_11_Handler	27	16+27: GPIO Port 0 pin 11 Handler
0x000000B0	PORT0_12_Handler	28	16+28: GPIO Port 0 pin 12 Handler
0x000000B4	PORT0_13_Handler	29	16+29: GPIO Port 0 pin 13 Handler

Address	Interrupt	Number	Description
0x000000B8	PORT0_14_Handler	30	16+30: GPIO Port 0 pin 14 Handler
0x000000BC	PORT0_15_Handler	31	16+31: GPIO Port 0 pin 15 Handler

4 通用异步收发器

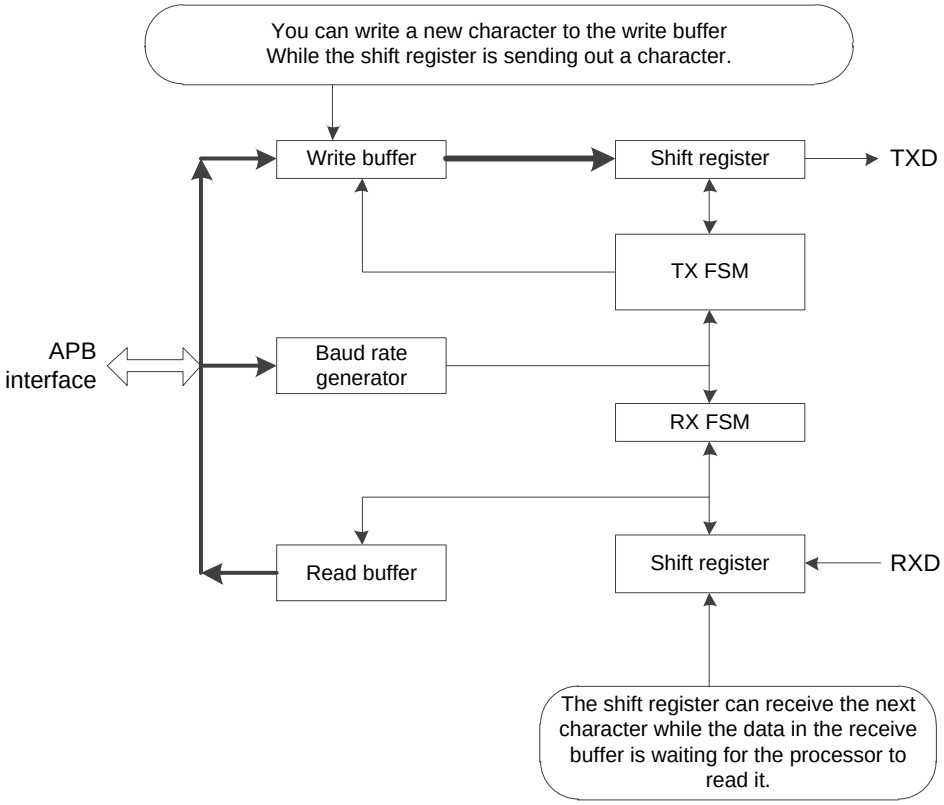
4.1 特征

Gowin_EMPU(GW1NS-4C)，包含 2 个通过 APB 总线访问的通用异步收发器 UART：

- 最大波特率为 921.6Kbit/s
- 无奇偶校验位
- 8 位数据位
- 1 位停止位

UART Buffering 如图 4-1 所示。

图 4-1 UART Buffering



UART 支持高速测试模式 HSTM（High Speed Test Mode），当寄存器 CTRL[6]设置为 1 时，串行数据每个周期传输 1 位，可以在很短时间内传输文本信息。

用户在使能 UART 时，必须设置波特率分频寄存器，例如，如果 APB1 总线频率运行在 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

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
DATA	0x000	RW	8	0x--	[7:0] Data Value
STATE	0x004	RW	4	0x0	[3] RX buffer overrun, write 1 to clear [2] TX buffer overrun, write 1 to clear [1] RX buffer full, read-only [0] TX buffer full, read-only
CTRL	0x008	RW	7	0x00	[6] High speed test mode for TX only

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[5] RX overrun interrupt enable [4] TX overrun interrupt enable [3] RX interrupt enable [2] TX interrupt enable [1] RX enable [0] TX enable
INTSTATU/INTCLEAR	0x00C	RW	4	0x0	[3] RX overrun interrupt, write 1 to clear [2] TX overrun interrupt, write 1 to clear [1] RX interrupt, write 1 to clear [0] TX interrupt, write 1 to clear
BAUDDIV	0x010	RW	20	0x00000	[19:0] Baud rate divider, the minimum number is 16

4.3 初始化定义

UART 初始化定义，如表 4-2 所示。

表 4-2 UART 初始化定义

名称	类型	数值	描述
UART_BaudRate	uint32_t	Max 921.6Kbit/s	Baud rate
UART_Mode	UARTMode_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX mode
UART_Int	UARTInt_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX interrupt
UART_Ovr	UARTOvr_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX overrun interrupt
UART_Hstm	FunctionalState	ENABLE/DISABLE	Enable/Disable TX high speed test mode

4.4 驱动程序使用方法

UART 驱动程序使用方法，如表 4-3 所示。

表 4-3 UART 驱动程序使用方法

名称	描述
UART_Init	Initializes UARTx
UART_GetRxBufferFull	Returns UARTx RX buffer full status
UART_GetTxBufferFull	Returns UARTx TX buffer full status

名称	描述
UART_GetRxBufferOverrunStatus	Returns UARTx RX buffer overrun status
UART_GetTxBufferOverrunStatus	Returns UARTx TX buffer overrun status
UART_ClearRxBufferOverrunStatus	Clears Rx buffer overrun status
UART_ClearTxBufferOverrunStatus	Clears Tx buffer overrun status
UART_SendChar	Sends a character to UARTx TX buffer
UART_SendString	Sends a string to UARTx TX buffer
UART_ReceiveChar	Receives a character from UARTx RX buffer
UART_GetBaudDivider	Returns UARTx baud rate divider value
UART_GetTxIRQStatus	Returns UARTx TX interrupt status
UART_GetRxIRQStatus	Returns UARTx RX interrupt status
UART_ClearTxIRQ	Clears UARTx TX interrupt status
UART_ClearRxIRQ	Clears UARTx RX interrupt status
UART_GetTxOverrunIRQStatus	Returns UARTx TX overrun interrupt status
UART_GetRxOverrunIRQStatus	Returns UARTx RX overrun interrupt status
UART_ClearTxOverrunIRQ	Clears UARTx TX overrun interrupt request
UART_ClearRxOverrunIRQ	Clears UARTx RX overrun interrupt request
UART_SetHSTM	Sets UARTx TX high speed test mode
UART_ClrHSTM	Clears UARTx TX high speed test mode

4.5 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 UART 软件编程参考设计，点击此链接获取如下参考设计：
cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign \Keil_RefDesign\uart
- Gowin_EMPU\ref_design\MCU_RefDesign \Keil_RefDesign\retarget
- Gowin_EMPU\ref_design\MCU_RefDesign \Keil_RefDesign\uart0_int
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_uart
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_retarget
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_uart0_int

5 定时器

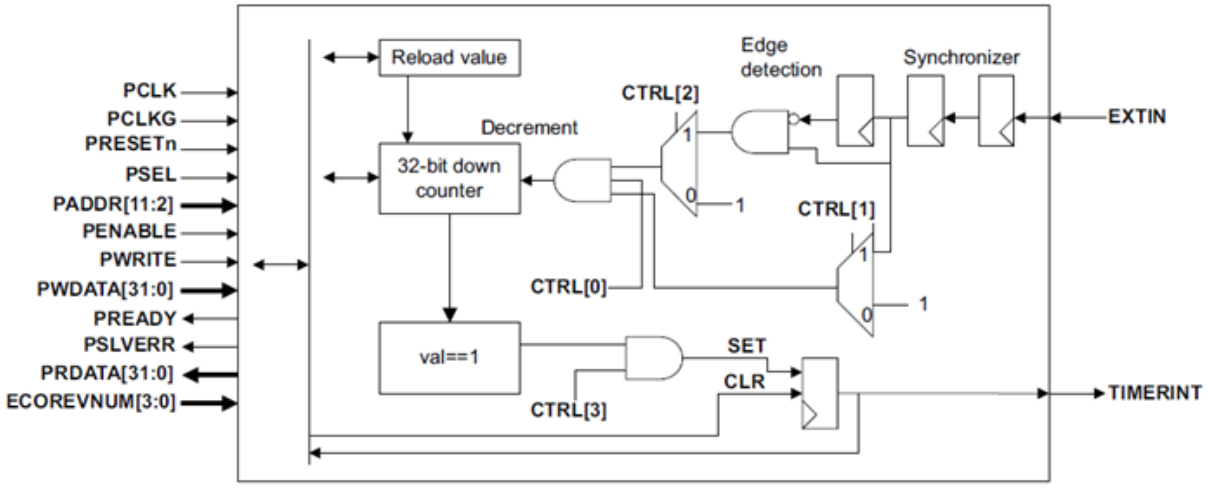
5.1 特征

Gowin_EMPU(GW1NS-4C)，包含 2 个通过 APB 总线访问的同步标准定时器：

- 32 位计数器
- 可以产生中断请求信号
- 可以使用外部输入信号 EXTIN 使能时钟
- TIMER0: EXTIN 连接 GPIO[1]
- TIMER1: EXTIN 连接 GPIO[6]

TIMER 结构如图 5-1 所示。

图 5-1 TIMER



5.2 寄存器定义

TIMER 寄存器定义，如表 5-1 所示。

表 5-1 TIMER 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CTRL	0x000	RW	4	0x0	[3] Timer interrupt enable [2] Select external input as clock [1] Select external input as enable [0] Enable
VALUE	0x004	RW	32	0x00000000	[31:0] Current value
RELOAD	0x008	RW	32	0x00000000	[31:0] Reload value, writing to this register sets the current value
INTSTATUS/INTCLEAR	0x00C	RW	1	0x0	[0] Timer interrupt, write 1 to clear

5.3 初始化定义

TIMER 初始化定义，如表 5-2 所示。

表 5-2 TIMER 初始化定义

名称	类型	数值	描述
Reload	uint32_t	-	Reload value
TIMER_Int	TIMERInt_TypeDef	SET/RESET	Enable/Disable interrupt
TIMER_Exti	TIMERExti_TypeDef	-	External input as enable or clock

5.4 驱动程序使用方法

TIMER 驱动程序使用方法，如表 5-3 所示。

表 5-3 TIMER 驱动程序使用方法

名称	描述
TIMER_Init	Initializes TIMERx
TIMER_StartTimer	Starts TIMERx
TIMER_StopTimer	Stops TIMERx
TIMER_GetIRQStatus	Returns TIMERx interrupt status
TIMER_ClearIRQ	Clears TIMERx interrupt status
TIMER_GetReload	Returns TIMERx reload value
TIMER_SetReload	Sets TIMERx reload value
TIMER_GetValue	Returns TIMERx current value
TIMER_SetValue	Sets TIMERx current value
TIMER_EnableIRQ	Enable TIMERx interrupt request

名称	描述
TIMER_DisableIRQ	Disable TIMERx interrupt request

5.5 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 Timer 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\timer
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_timer

6 看门狗

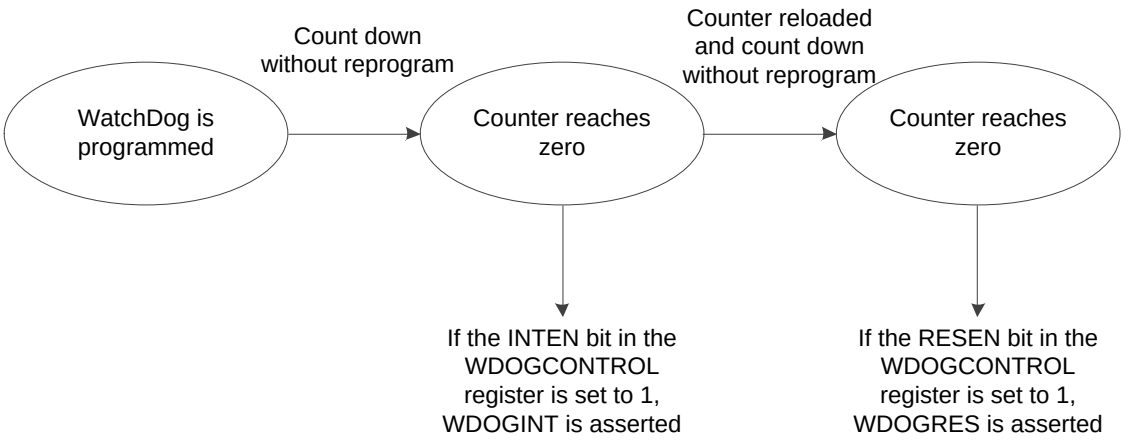
6.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 APB 总线访问的看门狗 WatchDog：

- 基于由 LOAD 寄存器初始化的 32 位逐减计数器
- 产生中断请求
- 当时钟使能，由 WDOGCLK 信号上升沿触发计数器递减
- 监视中断，当计数器递减到 0 时，产生复位请求，计数器停止
- 响应软件崩溃引起的复位，提供软件恢复方法

WatchDog 操作如图 6-1 所示。

图 6-1 WatchDog Operation



6.2 寄存器定义

WatchDog 寄存器定义，如表 6-1 所示。

表 6-1 WatchDog 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
LOAD	0x00	RW	32	0xFFFFFFFF	The value from which the counter is to decrement
VALUE	0x04	RO	32	0xFFFFFFFF	The current value of the decrementing counter
CTRL	0x08	RW	2	0x0	[1] Enable reset output [0] Enable the interrupt
INTCLR	0x0C	WO	-	-	Clear the watchdog interrupt and reloads the counter
RIS	0x10	RO	1	0x0	Raw interrupt status from the counter
MIS	0x14	RO	1	0x0	Enable interrupt status from the counter
RESERVED	0xC00-0x014	-	-	-	Reserved
LOCK	0xC00	RW	32	0x00000000	[32:1] Enable register writes [0] Register write enable status
RESERVED	0xF00-0xC00	-	-	-	Reserved
ITCR	0xF00	RW	1	0x0	Integration test mode enable
ITOP	0xF04	WO	2	0x0	[1] Integration test WDOGRES value [0] Integration test WDOGINT value

6.3 初始化定义

WatchDog 初始化定义，如表 6-2 所示。

表 6-2 WatchDog 初始化定义

名称	类型	数值	描述
WDOG_Reload	uint32_t	-	Reload value
WDOG_Lock	WDOGLock_TypeDef	SET/RESET	Enable/Disable lock register write access
WDOG_Res	WDOGRes_TypeDef	SET/RESET	Enable/Disable reset flag
WDOG_Int	WDOGInt_TypeDef	SET/RESET	Enable/Disable interrupt flag
WDOG_ITMode	WDOGMode_Typedef	SET/RESET	Enable/Disable integration test mode flag

6.4 驱动程序使用方法

WatchDog 驱动程序使用方法，如表 6-3 所示。

表 6-3 WatchDog 驱动程序使用方法

名称	描述
WDOG_Init	Initializes WatchDog
WDOG_RestartCounter	Restart watchdog counter
WDOG_GetCounterValue	Returns counter value
WDOG_SetResetEnable	Sets reset enable
WDOG_GetResStatus	Returns reset status
WDOG_SetIntEnable	Sets interrupt enable
WDOG_GetIntStatus	Returns interrupt enable
WDOG_ClrIntEnable	Clears interrupt enable
WDOG_GetRawIntStatus	Returns raw interrupt status
WDOG_GetMaskIntStatus	Returns masked interrupt status
WDOG_LockWriteAccess	Disable write access all registers
WDOG_UnlockWriteAccess	Enable write access all registers
WDOG_SetITModeEnable	Sets integration test mode enable
WDOG_ClrITModeEnable	Clears integration test mode enable
WDOG_GetITModeStatus	Returns integration test mode status
WDOG_SetITOP	Sets integration test output reset or interrupt
WDOG_GetITOPResStatus	Returns integration test output reset status
WDOG_GetITOPIntStatus	Returns integration test output interrupt status
WDOG_ClrITOP	Clears integration test output reset or interrupt

6.5 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 WatchDog 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\wdog
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_wdog

7 通用输入输出

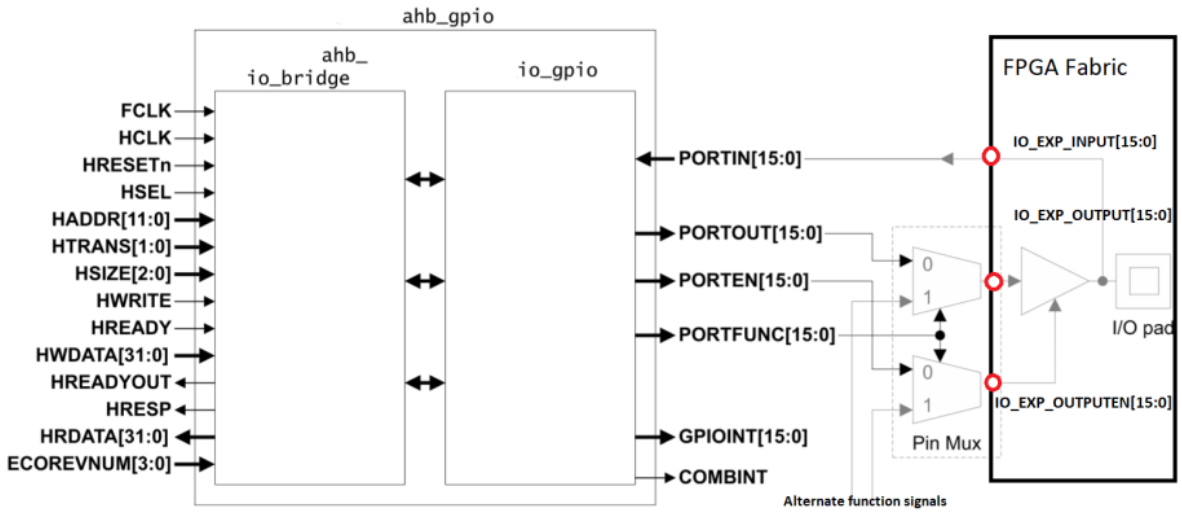
7.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 AHB 总线访问的 16 位输入输出接口的 GPIO 模块：

- 与 FPGA 内核系统连接；
- 每个 IO 管脚可以产生中断；
- 支持位掩码；
- 管脚复用功能。

GPIO 架构如图 7-1 所示。

图 7-1 GPIO Block



7.2 寄存器定义

GPIO 寄存器定义，如表 7-1 所示。

表 7-1 GPIO 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DATA	0x0000	RW	16	0x----	[15:0] Data value Read Sampled at pin Write to data output register Read back value goes through double flip-flop synchronization logic with delay of two cycles
DATAOUT	0x0004	RW	16	0x0000	[15:0] Data output register value Read current value of data output register write to data output register
RESERVED	0x0008 -0x000C	-	-	-	Reserved
OUTENSET	0x0010	RW	16	0x0000	[15:0] Output enable set Write 1 to set the output enable bit Write 0 no effect Read back 0 indicates the signal direction as input 1 indicates the signal direction as output
OUTENCLR	0x0014	RW	16	0x0000	[15:0] Output enable clear Write 1 to clear the output enable bit Write 0 no effect Read back 0 indicates the signal direction as input 1 indicates the signal direction as output
ALTFUNCSET	0x0018	RW	16	0x0000	[15:0] Alternative function set Write 1 to set the ALTFUNC bit Write 0 no effect Read back 0 for I/O 1 for an alternate function
ALTFUNCCLR	0x001C	RW	16	0x0000	[15:0] Alternative function clear Write 1 to clear the ALTFUNC bit Write 0 no effect Read back 0 for I/O

寄存器名称	地址偏移	类型	宽度	初始值	描述
					1 for an alternate function
INTENSET	0x0020	RW	16	0x0000	[15:0] Interrupt enable set Write 1 to set the enable bit Write 0 no effect Read back 0 indicates interrupt disabled 1 indicates interrupt enabled
INTENCLR	0x0024	RW	16	0x0000	[15:0] Interrupt enable clear Write 1 to clear the enable bit Write 0 no effect Read back 0 indicates interrupt disabled 1 indicates interrupt enabled
INTTYPESET	0x0028	RW	16	0x0000	[15:0] Interrupt type set Write 1 to set the interrupt type bit Write 0 no effect Read back 0 for LOW/HIGH level 1 for falling edge or rising edge
INTTYPECLR	0x002C	RW	16	0x0000	[15:0] Interrupt type clear Write 1 to clear the interrupt type bit Write 0 no effect Read back 0 for LOW/HIGH level 1 for falling edge or rising edge
INTPOLSET	0x0030	RW	16	0x0000	[15:0] Polarity-level, edge IRQ configuration Write 1 to set the interrupt polarity bit Write 0 no effect Read back 0 for LOW level or falling edge 1 for HIGH level or rising edge
INTPOLCLR	0x0034	RW	16	0x0000	[15:0] Polarity-level, edge IRQ configuration Write 1 to clear the interrupt polarity bit Write 0 no effect Read back 0 for LOW level or falling edge 1 for HIGH level or rising edge
INTSTATUS/INTCLEAR	0x0038	RW	16	0x0000	[15:0] Write IRQ status clear register Write 1 to clear interrupt request Write 0 no effect Read back IRQ status register
MASKLOW	0x0400	RW	16	0x----	Lower 8-bits masked access

寄存器名称	地址偏移	类型	宽度	初始值	描述
BYTE	-0x07FC				[9:2] of the address value are used as enable bit mask for the access [15:8] not used [7:0] Data for lower byte access, with [9:2] of address value used as enable mask for each bit
MASKHIGHBYTE	0x0800 -0x0BFC	RW	16	0x----	Higher 8-bits masked access [9:2] of the address value are used as enable bit mask for the access [15:8] Data for higher byte access, with [9:2] of address value used as enable mask for each bit [7:0] not used
RESERVED	0x0C00 -0x0FCF	-	-	-	Reserved

7.3 初始化定义

GPIO 初始化定义，如表 7-2 所示。

表 7-2 GPIO 初始化定义

名称	类型	数值	描述
GPIO_Pin	uint32_t	GPIO_Pin_0 GPIO_Pin_1 GPIO_Pin_2 GPIO_Pin_3 GPIO_Pin_4 GPIO_Pin_5 GPIO_Pin_6 GPIO_Pin_7 GPIO_Pin_8 GPIO_Pin_9 GPIO_Pin_10 GPIO_Pin_11 GPIO_Pin_12 GPIO_Pin_13 GPIO_Pin_14 GPIO_Pin_15	16 bits GPIO Pins
GPIO_Mode	GPIOMode_TypeDef	GPIO_Mode_IN	16 bits GPIO Pins mode

名称	类型	数值	描述
		GPIO_Mode_OUT GPIO_Mode_AF	
GPIO_Int	GPIOInt_TypeDef	GPIO_Int_Disable GPIO_Int_Low_Level GPIO_Int_High_Level GPIO_Int_Falling_Edge GPIO_Int_Rising_Edge	16 bits GPIO Pins interrupt

7.4 驱动程序使用方法

GPIO 驱动程序使用方法，如表 7-3 所示。

表 7-3 GPIO 驱动程序使用方法

名称	描述
GPIO_Init	Initializes GPIOx
GPIO_SetOutEnable	Sets GPIOx output enable
GPIO_ClrOutEnable	Clears GPIOx output enable
GPIO_GetOutEnable	Returns GPIOx output enable
GPIO_SetBit	GPIO output one
GPIO_ResetBit	GPIO output zero
GPIO_WriteBits	GPIO output
GPIO_ReadBits	GPIO input
GPIO_SetAltFunc	Sets GPIOx alternate function enable
GPIO_ClrAltFunc	Clears GPIOx alternate function enable
GPIO_GetAltFunc	Returns GPIOx alternate function enable
GPIO_IntClear	Clears GPIOx interrupt request
GPIO_GetIntStatus	Returns GPIOx interrupt status
GPIO_SetIntEnable	Sets GPIOx interrupt enable Returns GPIOx interrupt status
GPIO_ClrIntEnable	Clears GPIOx interrupt enable Returns GPIOx interrupt enable
GPIO_SetIntHighLevel	Setups GPIOx interrupt as high level
GPIO_SetIntRisingEdge	Setups GPIOx interrupt as rising edge
GPIO_SetIntLowLevel	Setups GPIOx interrupt as low level
GPIO_SetIntFallingEdge	Setups GPIOx interrupt as falling edge

名称	描述
GPIO_MaskedWrite	Setups GPIOx output value using masked access

7.5 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 GPIO 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\led
- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\keyscan
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_led
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_keyscan

8 实时时钟

8.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 APB 总线访问的 32 位实时时钟 RTC 模块：

- APB 总线接口
- 32-bit 计数器
- 32-bit Match 寄存器
- 32-bit 比较器

MCU 通过 APB 总线接口与 RTC 读写数据、控制和读取状态信息。在连续输入时钟 CLK1HZ（端口 rtc_src_clk 接入 3.072MHz 时钟输入，RTC 内部分频为 1Hz）上升沿时，32-bit 计数器递增。

此计数器是不同步计数器，不可重载。在系统复位时，此计数器从 1 开始计数，递增到最大值 0xFFFFFFFF，然后回绕到 0 开始继续递增。

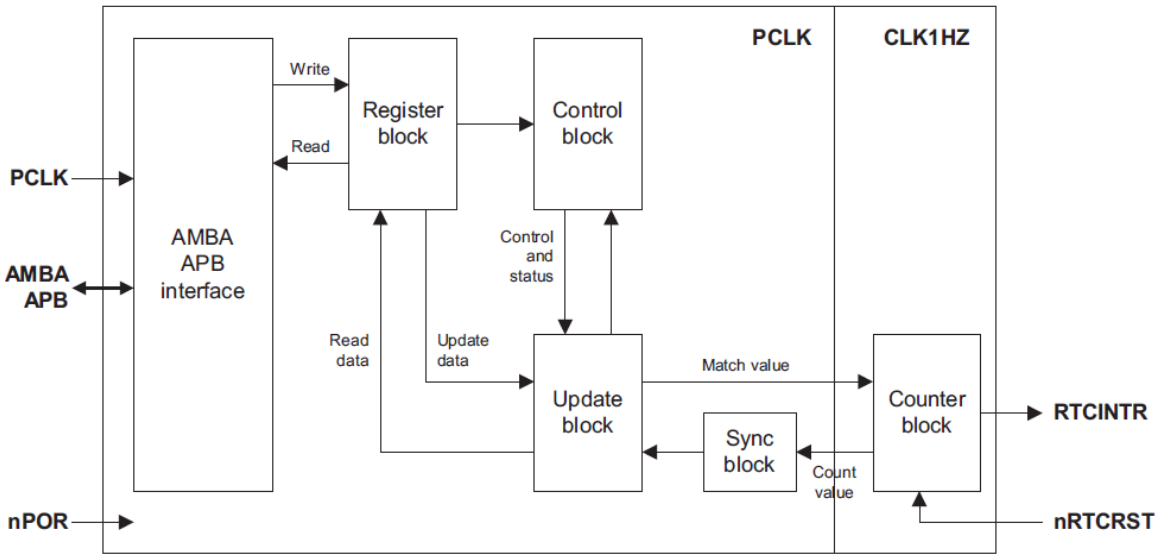
通过写 Load 寄存器 RTC_LOAD_VALUE，实现 RTC 加载或更新。

通过读 Data 寄存器 RTC_CURRENT_DATA，获取 RTC 当前时钟。

通过写 RTC_MATCH_VALUE 寄存器，编程 Match 寄存器。

RTC 结构如图 8-1 所示。

图 8-1 RTC Block



8.2 寄存器定义

RTC 寄存器定义，如表 8-1 所示。

表 8-1 RTC 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RTC_CURRENT_DATA	0x000	RO	32	0x00000000	Data Register [31:0] Current value
RTC_MATCH_VALUE	0x004	RW	32	0x00000000	Match Register If current value equals match register's value, generate interrupt [31:0] Match data
RTC_LOAD_VALUE	0x008	RW	32	0x00000000	Load Register Initialized value, start counter based on this value [31:0] Load data
RTC_CTROLLER_REG	0x00C	RW	32	0x00000000	Control Register Start RTC counter [31:1] Reserved [0] Start RTC counter
RTC_IMSC	0x010	RW	32	0x00000000	Interrupt mask set and clear register Enable or disable interrupt [31:1] Reserved [0] Enable interrupt
RTC_RIS	0x014	RO	32	0x00000000	Raw interrupt status register

寄存器名称	地址偏移	类型	宽度	初始值	描述
					Get current raw unmasked interrupt status [31:1] Reserved [0] Current raw unmasked interrupt status
RTC_MIS	0x018	RO	32	0x00000000	Masked interrupt status register Get current masked interrupt status [31:1] Reserved [0] Current masked interrupt status
RTC_INTR_CLEAR	0x01C	WO	32	0x00000000	Interrupt clear register Clear current interrupt [31:1] Reserved [0] Clear current interrupt

8.3 驱动程序使用方法

RTC 驱动程序使用方法，如表 8-2 所示。

表 8-2 RTC 驱动程序使用方法

名称	描述
RTC_init	Initializes RTC
Get_Current_Value	Gets RTC current value of data register
Set_Match_Value	Sets RTC match value of match register
Get_Match_Value	Gets RTC match value of match register
Set_Load_Value	Sets RTC load value of load register
Get_Load_Value	Gets RTC load value of load register
Start_RTC	Starts RTC counter
Close_RTC	Closes RTC counter
RTC_Inter_Mask_Set	Sets RTC interrupt mask
Get_RTC_Control_value	Gets value of control register
RTC_Inter_Mask_Clr	Clears RTC interrupt mask
Get_RTC_Inter_Mask_value	Gets RTC interrupt mask
Clear_RTC_interrupt	Clears RTC interrupt

8.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 RTC 软件编程参考设计，点击此链接获取如下参考设计：
cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\rtc
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_rtc

9 串行外设接口

9.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 APB 总线访问的串行外设接口 SPI Master 模块：

- APB 总线接口
- 全双工同步串行数据传输
- 支持 Master 工作模式
- 支持可配置的时钟极性和相位
- SPI 产生的串行时钟频率可配置
- 数据接收寄存器和数据发送寄存器 8 位宽

9.2 寄存器定义

SPI Master 寄存器定义，如表 9-1 所示。

表 9-1 SPI Master 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RDATA	0x00	RO	8	0x00	Reads data register
WDATA	0x04	WO	8	0x00	Writes data register
STATUS	0x08	RW	8	0x00	[7] Error status [6] Receives ready status [5] Transmits ready status [4] Be transmitting [3] Transmits overrun error status [2] Receives overrun error status [1:0] Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
SSMASK	0x0C	RW	8	0x00	Unused selected slave address
CTRL	0x10	RW	5	0x00	[4:3] Clock selected [2] Polarity [1] Phase [0] Direction

9.3 初始化定义

SPI Master 初始化定义，如表 9-2 所示。

表 9-2 SPI Master 初始化定义

名称	类型	数值	描述
DIRECTION	uint8_t	1/0	MSB/LSB first transmission 0: MSB first; 1: LSB first.
PHASE	uint8_t	1/0	Posedge/Negedge transmit data 0: Sample at posedge edge; 1: Sample at negedge edge.
POLARITY	uint8_t	1/0	Initialize polarity to one/zero 0: Idle sclk low; 1: Idle sclk high.
CLKSEL	uint8_t	CLKSEL_CLK_DIV_2 CLKSEL_CLK_DIV_4 CLKSEL_CLK_DIV_6 CLKSEL_CLK_DIV_8	Select clock divided 2/4/6/8

9.4 驱动程序使用方法

SPI Master 驱动程序使用方法，如表 9-3 所示。

表 9-3 SPI Master 驱动程序使用方法

名称	描述
SPI_Init	Initializes SPI
SPI_SetDirection	Sets direction
SPI_ClrDirection	Clears direction
SPI_GetDirection	Returns direction
SPI_SetPhase	Sets phase

名称	描述
SPI_ClrPhase	Clears phase
SPI_GetPhase	Returns phase
SPI_SetPolarity	Sets polarity
SPI_ClrPolarity	Clears polarity
SPI_GetPolarity	Returns polarity
SPI_SetClkSel	Sets clock selection
SPI_GetClkSel	Returns clock selection
SPI_GetToeStatus	Reads transmit overrun error status
SPI_GetRoeStatus	Reads receive overrun error status
SPI_GetTmtStatus	Reads transmitting status
SPI_GetTrdyStatu	Reads transmit ready status
SPI_GetRrdyStatus	Reads receive ready error status
SPI_GetErrStatus	Reads error status
SPI_ClrToeStatus	Clears transmit overrun error status
SPI_ClrRoeStatus	Clear receive overrun error status
SPI_ClrErrStatus	Clears error status
SPI_ReadWriteByte	Full duplex read and write a byte
SPI_WriteData	Writes data
SPI_ReadData	Reads data
SPI_Select_Slave	Selects slave

9.5 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 SPI Master 软件编程参考设计，点击此链接获取如下参考设计：
cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_deisgn\MCU_RefDesign\Keil_RefDesign\spi
- Gowin_EMPU\ref_deisgn\MCU_RefDesign\GMD_RefDesign\cm3_spi

10 系统控制器

10.1 寄存器定义

SYSCON 寄存器定义，如表 10-1 所示。

表 10-1 SYSCON 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
REMAP	0x000	RW	1	0x0	Remap control register
PMUCTRL	0x004	RW	1	0x0	PMU control register
RESETOP	0x008	RW	1	0x0	reset option register
RESERVED0	0x00C	-	-	-	Reserved
RSTINFO	0x010	RW	3	0x0	[2] Lockup reset [1] Watchdog reset request [0] System reset request

10.2 驱动程序使用方法

SYSCON 驱动程序使用方法，如表 10-2 所示。

表 10-2 SYSCON 驱动程序使用方法

名称	描述
SYSCON_Init	Initializes SYSCON
SYSCON_GetRemap	Returns REMAP
SYSCON_GetPmuctrlEnable	Returns PMUCTRL Enable
SYSCON_GetResetopLockuprst	Returns RESETOP LOCKUPRST
SYSCON_GetRstinfoSysresetreq	Returns RSTINFO SYSRESETREQ
SYSCON_GetRstinfoWdogresetreq	Returns RSTINFO SYSRESETREQ
SYSCON_GetRstinfoLockreset	Returns RSTINFO SYSRESETREQ

11 内部集成电路总线

11.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 APB 总线访问的内部集成电路总线 I²C Master 模块：

- APB 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

11.2 寄存器定义

I²C Master 寄存器定义，如表 11-1 所示。

表 11-1 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	16	0xFFFF	Clock prescale register [15:0] Prescale value = sys_clk/(5*SCL)-1
CTR	0x04	RW	8	0x00	[7] Enable I²C function [6] Enable I²C interrupt [5:0] Reserved
TXR	0x08	WO	8	0x00	[7:1] Next transmission data [0] Data direction

寄存器名称	地址偏移	类型	宽度	初始值	描述
RXR	0x0C	RO	8	0x00	[7:0] Last received data
CR	0x010	WO	8	0x00	[7] Start transmission status [6] Over transmission status [5] Read enable, read data from slave [4] Write enable, write data to slave [3] Acknowledge [2:1] Reserved [0] Interrupt acknowledge
SR	0x14	RO	8	0x00	[7] Receive acknowledge signal from slave [6] I2C busy status [5] Arbitration loss [4:2] Reserved [1] Data transmission status flag [0] Interrupt flag

11.3 驱动程序使用方法

I²C Master 驱动程序使用方法，如表 11-2 所示。

表 11-2 I²C Master 驱动程序使用方法

名称	描述
I2C_Init	I²C Initialization
I2C_SendByte	Sends a byte to I²C bus
I2C_SendBytes	Sends multiple bytes to I²C bus
I2C_SendData	Sends multiple bytes to I²C bus once time
I2C_ReceiveByte	Reads a byte from I²C bus
I2C_ReadBytes	Reads multiple bytes from I²C bus
I2C_ReceiveData	Reads multiple bytes from I²C bus once time
I2C_Rate_Set	Sets I²C traffic rate
I2C_Enable	Enable I²C bus
I2C_UnEnable	Disable I²C bus
I2C_InterruptOpen	Opens I²C interrupt
I2C_InterruptClose	Closes I²C interrupt

11.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 I^2C Master 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\i2c
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_i2c

12 SysTick

12.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个 24 位系统节拍的内核定时器 SysTick，具有自动重载和溢出中断功能，可以通过此定时器获得一定的时间间隔。

12.2 寄存器定义

SysTick 寄存器定义，如表 12-1 所示。

表 12-1 SysTick 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CTRL	0x00	RW	17	0x00000	[16] Counts down to zero flag [2] External clock source [1] Enable interrupt request [0] Enable SysTick
LOAD	0x04	RW	24	0x000000	[23:0] Reloads value
VAL	0x08	RW	24	0x000000	[23:0] Returns current count down value
CALIB	0x0C	RW	32	0x00000000	[31] Whether a separate reference clock is provided [30] Whether the TENMS value is exact [23:0] 10ms calibration value

12.3 驱动程序使用方法

SysTick 驱动程序使用方法，如表 12-2 所示。

表 12-2 SysTick 驱动程序使用方法

名称	描述
SysTick_Config	Initializes and starts the SysTick counter and interrupt

12.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 SysTick 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\systick
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_systick

13 内存管理

13.1 特征

Gowin_EMPU(GW1NS-4C)，支持动态内存管理方法，通过重定向定义内存申请 malloc 与内存释放 free 函数，实现动态申请与释放内存。

13.2 驱动程序使用方法

内存管理驱动程序使用方法，如表 13-1 所示。

表 13-1 内存管理驱动程序使用方法

名称	描述
mem_init	Initializes memory management
mymemset	Sets the values of memory space
mymemcpy	Copies from source to destination
mymemcmp	Compares source with destination
mymalloc	Allocates a memory space dynamically
myfree	Frees a memory space

13.3 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的内存管理软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\mm
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_mm

14 SPI_Nor_Flash

14.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 APB 总线访问的 SPI_Nor_Flash 模块：

- 串行非易失闪存
- 支持 APB 总线配置接口
- 支持标准 SPI 接口
- 支持读、写和擦除功能
- 支持 GW1NSR-4C/GW1NSER-4C QN48G 器件

14.2 寄存器定义

SPI_Nor_Flash 寄存器定义，如表 14-1 所示。

表 14-1 SPI_Nor_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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					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 lease significant bit first 0 = Most significant bit first 1 = Least significant bit first [2] SPI master/slave mode selection 0 = Master mode 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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[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 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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[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 [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)

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[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
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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[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 [8] 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

14.3 驱动程序使用方法

SPI_Nor_Flash 驱动程序使用方法，如表 14-2 所示。

表 14-2 SPI_Nor_Flash 驱动程序使用方法

名称	描述
spi_nor_flash_init	Initializes SPI_Nor_Flash
change_mode_spi_nor_flash	Switch SPI_Nor_Flash mode to read, write, erase memory
spi_nor_flash_read	Read data from SPI_Nor_Flash
spi_nor_flash_write	Write data into SPI_Nor_Flash
spi_nor_flash_write_read	Write data into SPI_Nor_Flash and read data from SPI_Nor_Flash once time
spi_nor_flash_page_program	Write data into SPI_Nor_Flash with pages
spi_nor_flash_sector_erase	Erase SPI_Nor_Flash with sector

14.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 SPI_Nor_Flash 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\spi_nor_flash
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_spi_nor_flash

15 PSRAM Memory Interface

15.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 AHB 总线访问的 PSRAM Memory Interface 模块：

- 支持 AHB 总线配置接口
- 与标准的 PSRAM 器件接口兼容
- 支持 GW1NSR-4C MG64P 器件
- 支持 8 位 PSRAM width
- 支持 16 位 DQ bus width
- 可编程突发长度 16、32、64、128
- 时钟比例为 1:2
- 支持初始延时为 6
- 支持固定延时模式
- 支持电源关闭选项
- 可配置的驱动强度
- 可配置的自刷新区域
- 可配置的刷新速率

15.2 寄存器定义

PSRAM Memory Interface 寄存器定义，如表 15-1 所示。

表 15-1 PSRAM Memory Interface 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CMD	0x00	RW	1	0x0	Command register [0] Operation type 0 = Read operation 1 = Write operation
ADDRESS	0x04	RW	21	0x0	Address register [20:0] Address of reading and writing data
WR_DATA0	0x08	RW	32	0x0	Write data register 0 [31:0] Write first 32bit data
WR_DATA1	0x0C	RW	32	0x0	Write data register 1 [31:0] Write second 32bit data
WR_DATA2	0x10	RW	32	0x0	Write data register 2 [31:0] Write third 32bit data
WR_DATA3	0x14	RW	32		Write data register 3 [31:0] Write fourth 32bit data
CMD_EN	0x18	WO	1		Command enable register [0] Enable PSRAM
READ_DONE	0x1C	RW	1		Read status register [0] Read done flag, auto set 1 if it is done, and need mcu to clear
RD_DATA0	0x20	RO	32		Read data register 0 [31:0] Read first 32bit data
RD_DATA1	0x24	RO	32		Read data register 1 [31:0] Read second 32bit data
RD_DATA2	0x28	RO	32		Read data register 2 [31:0] Read third 32bit data
RD_DATA3	0x2C	RO	32		Read data register 3 [31:0] Read fourth 32bit data
INTI_DONE	0x30	RO	1		Initialization done register [0] PSRAM hardware initialization

寄存器名称	地址偏移	类型	宽度	初始值	描述
					done flag 0 = Initialization failed 1 = Initialization done

15.3 驱动程序使用方法

PSRAM Memory Interface 驱动程序使用方法，如表 15-2 所示。

表 15-2 PSRAM Memory Interface 驱动程序使用方法

名称	描述
PSRAM_Check_Init_Status	Check the status of PSRAM initialization
PSRAM_Mode_Set	Set the mode for PSRAM write and read
PSRAM_Address_Set	Set the address of PSRAM and save data into this address
PSRAM_Read_Data_Buff	Read data from the buffer of PSRAM
PSRAM_Cmd_Enable	Enable the command of PSRAM
PSRAM_Read_Done_Flag	Get the flag of read PSRAM done
PSRAM_Clear_Read_Done_Flag	Clear the flag of read PSRAM done
PSRAM_Write_Data_Buff	Write data into the buffer of PSRAM
PSRAM_Cmd_Unable	Disable the command of PSRAM
PSRAM_Write_Data_Package	Write a package data into PSRAM
PSRAM_Read_Data_Package	Read a package data from PSRAM

15.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 PSRAM Memory Interface 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\psram
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_psram

16 HyperRAM Memory Interface

16.1 特征

Gowin_EMPU(GW1NS-4C)，包含 1 个通过 AHB 总线访问的 HyperRAM Memory Interface 模块：

- 支持 AHB 总线配置接口
- 与标准的 HyperRAM 器件接口兼容
- 支持 GW1NSR-4C/GW1NSER-4C QN48P 器件
- 支持 8 位 PSRAM Width
- 支持 8 位 DQ bus Width
- 可编程突发长度 16、32、64、128
- 时钟比例为 1:2
- 支持初始延时为 6
- 支持固定延时模式
- 支持电源关闭选项
- 可配置的驱动强度
- 可配置的自刷新区域
- 可配置的刷新速率

16.2 寄存器定义

HyperRAM Memory Interface 寄存器定义，如表 16-1 所示。

表 16-1 HyperRAM Memory Interface 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CMD	0x00	RW	1	0x0	Command register [0] Operation type 0 = Read operation 1 = Write operation
ADDRESS	0x04	RW	21	0x0	Address register [20:0] Address of reading and writing data
WR_DATA0	0x08	RW	32	0x0	Write data register 0 [31:0] Write first 32bit data
WR_DATA1	0x0C	RW	32	0x0	Write data register 1 [31:0] Write second 32bit data
WR_DATA2	0x10	RW	32	0x0	Write data register 2 [31:0] Write third 32bit data
WR_DATA3	0x14	RW	32		Write data register 3 [31:0] Write fourth 32bit data
CMD_EN	0x18	WO	1		Command enable register [0] Enable HyperRAM
READ_DONE	0x1C	RW	1		Read status register [0] Read done flag, auto set 1 if it is done, and need mcu to clear
RD_DATA0	0x20	RO	32		Read data register 0 [31:0] Read first 32bit data
RD_DATA1	0x24	RO	32		Read data register 1 [31:0] Read second 32bit data
RD_DATA2	0x28	RO	32		Read data register 2 [31:0] Read third 32bit data
RD_DATA3	0x2C	RO	32		Read data register 3 [31:0] Read fourth 32bit data
INTI_DONE	0x30	RO	1		Initialization done register [0] HyperRAM hardware initialization

寄存器名称	地址偏移	类型	宽度	初始值	描述
					done flag 0 = Initialization failed 1 = Initialization done

16.3 驱动程序使用方法

HyperRAM Memory Interface 驱动程序使用方法，如表 16-2 所示。

表 16-2 HyperRAM Memory Interface 驱动程序使用方法

名称	描述
HPRAM_Check_Init_Status	Check the status of HyperRAM initialization
HPRAM_Mode_Set	Set the mode for HyperRAM write and read
HPRAM_Address_Set	Set the address of HyperRAM and save data into this address
HPRAM_Read_Data_Buff	Read data from the buffer of HyperRAM
HPRAM_Cmd_Enable	Enable the command of HyperRAM
HPRAM_Read_Done_Flag	Get the flag of read HyperRAM done
HPRAM_Clear_Read_Done_Flag	Clear the flag of read HyperRAM done
HPRAM_Write_Data_Buff	Write data into the buffer of HyperRAM
HPRAM_Cmd_Unable	Disable the command of HyperRAM
HPRAM_Write_Data_Package	Write a package data into HyperRAM
HPRAM_Read_Data_Package	Read a package data from HyperRAM

16.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 HyperRAM Memory Interface 软件编程参考设计，点击此链接获取如下参考设计：cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\hyper_ram
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_hyper_ram

17 嵌入式实时操作系统

Gowin_EMPU(GW1NS-4C)，支持 uC/OS-III 和 FreeRTOS 嵌入式实时操作系统编程。

17.1 uC/OS-III

17.1.1 特征

- uC/OS-III 是一个可扩展的，可固化的，抢占式的实时内核，管理的任务个数不受限制
- uC/OS-III 是第三代内核，提供了现代实时内核所期望的功能，包括资源管理、同步、任务间通信等
- uC/OS-III 提供了很多其它实时内核所没有的特性，比如能在运行时测量运行性能，直接发送信号或消息给任务，任务能同时等待多个信号量和消息队列
- Gowin_EMPU(GW1NS-4C)已成功移植 uC/OS-III 参考设计
- uC/OS-III 源代码请在 Micrium 官网 http://www.micrium.com 下载

17.1.2 操作系统版本

Gowin_EMPU(GW1NS-4C)参考设计使用的 uC/OS-III 版本为 V3.03.00。

17.1.3 操作系统配置

- 用户可以通过修改 UCOSIII_CONFIG\os_cfg.h 和 os_cfg_app.h 来配置 uC/OS-III
- 用户可以通过修改 UCOS_BSP\bsp.c 和 bsp.h 来支持所用开发板

17.1.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 uC/OS-III 软件编程参考设计，点击此链接获取如下参考设计：
cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\ucos_iii
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_ucos_iii

17.2 FreeRTOS

17.2.1 特征

- FreeRTOS 是一个轻量级的实时操作系统。
- FreeRTOS 作为一个轻量级的操作系统，功能包括：任务管理、时间管理、信号量、消息队列、内存管理、记录功能、软件定时器、协程等，可基本满足较小系统的需要。
- FreeRTOS 操作系统是完全免费的操作系统，具有源码公开、可移植、可裁减、调度策略灵活的特点。
- Gowin_EMPU(GW1NS-4C)已成功移植 FreeRTOS 参考设计。
- FreeRTOS 源代码请在 FreeRTOS 官网 http://www.FreeRTOS.org 下载。

17.2.2 操作系统版本

Gowin_EMPU(GW1NS-4C)参考设计使用的 FreeRTOS 版本为 V10.2.1。

17.2.3 操作系统配置

用户可以通过修改 include\FreeRTOSConfig.h 来配置 FreeRTOS。

17.2.4 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK（V5.26 及以上版本）和 GOWIN MCU Designer（V1.1 及以上版本）软件环境的 FreeRTOS 软件编程参考设计，点击此链接获取如下参考设计：
cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip：

- Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign\free_rtos
- Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign\cm3_free_rtos


GOWIN高云
智慧逻辑 定制未来