




Gowin SPI Flash Interface IP 用户指南

IPUG1015-1.0, 2023-02-22

版权所有 © 2023 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin、GowinSynthesis、云源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
2023/02/22	1.0	初始版本。

目录

目录	i
图目录	iv
表目录	vi
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	1
1.4 技术支持与反馈	2
2 概述	3
2.1 概述	3
2.2 主要特征	3
2.3 资源利用	4
3 功能描述	5
3.1 系统框图	5
3.2 端口列表	7
3.3 参数配置	9
3.4 寄存器描述	9
3.4.1 SPI 传输格式寄存器(0x10)	10
3.4.2 SPI 传输控制寄存器(0x20)	10
3.4.3 SPI 命令寄存器(0x24)	12
3.4.4 SPI 地址寄存器(0x28)	12
3.4.5 SPI 数据寄存器(0x2C)	13
3.4.6 SPI 控制寄存器(0x30)	13
3.4.7 SPI 状态寄存器(0x34)	13
3.4.8 SPI 中断使能寄存器(0x38)	14
3.4.9 SPI 中断状态寄存器(0x3C)	14
3.4.10 SPI 接口时序寄存器(0x40)	14

3.4.11 配置寄存器(0x7C)	15
3.5 硬件配置选项	15
3.5.1 TX FIFO 深度	15
3.5.2 RX FIFO 深度	15
3.5.3 Memory-Mapped 读取	15
3.5.4 SPI 接口时钟参数	15
3.6 读写操作流程	15
3.6.1 Write Enable (06H)操作过程.....	16
3.6.2 Write Disable (04H)操作过程.....	17
3.6.3 Read Status Register (05H/35H)操作过程	17
3.6.4 Write Status Register (01H)操作过程	18
3.6.5 Read Data (03H)操作过程	18
3.6.6 Page Program (02H)操作过程	19
3.6.7 Sector Erase (20H)操作过程	21
3.6.8 32 KB Block Erase (52H)操作过程	22
3.6.9 64 KB Block Erase (D8H)操作过程.....	22
3.6.10 Chip Erase (60/C7H)操作过程.....	23
3.6.11 Read Manufacturer/Device ID (90H)操作过程.....	24
3.6.12 Read JEDEC ID (9FH)操作过程	24
3.6.13 Read Unique ID (4BH)操作过程	25
3.6.14 Fast Read Dual Output (3BH)操作过程.....	26
3.6.15 Fast Read Dual I/O (BBH)操作过程.....	27
3.6.16 Read Manufacturer/Device ID Dual I/O (92H)操作过程	28
3.6.17 Quad Page Program (32H)操作过程	29
3.6.18 Fast Read Quad Output (6BH)操作过程	31
3.6.19 Fast Read Quad I/O (EBH)操作过程	32
3.6.20 Read Manufacturer/Device ID Quad I/O (94H)操作过程.....	32
3.7 时序说明.....	33
3.7.1 SPI 接口时序.....	33
3.7.2 AHB 接口时序	34
3.7.3 APB 接口时序	35
4 界面配置	37
5 参考设计	40
6 文件交付	41
6.1 文档.....	41
6.2 设计源代码（加密）	41

6.3 参考设计..... 41

图目录

图 3-1 系统框图	6
图 3-2 Gowin SPI Flash Interface IP IO 端口示意图	7
图 3-3 Write Enable 时序图	17
图 3-4 Write Disable 时序图	17
图 3-5 Read Status Register 时序图	18
图 3-6 Write Status Register 时序图.....	18
图 3-7 Read Data 时序图	19
图 3-8 Page Program 时序图	21
图 3-9 Sector Erase 时序图.....	21
图 3-10 32KB Block Erase 时序图.....	22
图 3-11 64KB Block Erase 时序图.....	23
图 3-12 Chip Erase 时序图.....	23
图 3-13 Read Manufacture ID/Device ID 时序图.....	24
图 3-14 Read JEDEC ID 时序图.....	25
图 3-15 Read Unique ID 时序图.....	26
图 3-16 Fast Read Dual Output 时序图.....	27
图 3-17 Fast Read Dual I/O 时序图.....	28
图 3-18 Read Manufacturer/Device ID Dual I/O 时序图.....	29
图 3-19 Quad Page Program 时序图.....	30
图 3-20 Fast Read Quad Output 时序图	31
图 3-21 Fast Read Quad I/O 时序图	32
图 3-22 Read Manufacturer/Device ID Quad I/O 时序图	33
图 3-23 SPI Flash 读接口时序示意图	34
图 3-24 SPI Flash 写接口时序图	34
图 3-25 AHB 总线无等待状态传输接口时序	34
图 3-26 AHB 总线有等待状态传输接口时序	35
图 3-27 APB 总线写接口时序	35
图 3-28 APB 总线读接口时序	36
图 4-1 打开 IP Core Generator.....	37

图 4-2 打开 SPI Flash Interface IP 核.....	38
图 4-3 SPI Flash Interface IP 核端口示意图.....	38
图 4-4 基本信息配置界面.....	39
图 4-5 Options 选项卡	39
图 5-1 参考设计基本结构图	40

表目录

表 1-1 术语、缩略语	1
表 2-1 Gowin SPI Flash Interface IP	3
表 2-2 占用资源	4
表 3-1 Gowin SPI Flash Interface IP 的端口列表	7
表 3-2 Gowin SPI Flash Interface IP 参数	9
表 3-3 寄存器列表	9
表 3-4 SPI 传输格式寄存器	10
表 3-5 SPI 传输控制寄存器	10
表 3-6 普通传输格式 Dummy 周期设置	12
表 3-7 SPI 命令寄存器	12
表 3-8 SPI 地址寄存器	12
表 3-9 SPI 数据寄存器	13
表 3-10 SPI 控制寄存器	13
表 3-11 SPI 状态寄存器	13
表 3-12 SPI 中断使能寄存器	14
表 3-13 SPI 中断状态寄存器	14
表 3-14 SPI 接口时序寄存器	14
表 3-15 配置寄存器	15
表 3-16 SPI Flash 部分命令列表	16
表 6-1 文档列表	41
表 6-2 SPI Flash Interface 设计源代码列表	41
表 6-3 RefDesign 文件夹内容列表	41

1 关于本手册

1.1 手册内容

Gowin® SPI Flash Interface IP 用户指南主要内容包括功能简介、端口说明、时序说明、配置调用、参考设计等，旨在帮助用户快速了解 Gowin SPI Flash Interface IP 的特性及使用方法。

1.2 相关文档

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

- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS117, GW1NR 系列 FPGA 产品数据手册](#)
- [DS891, GW1NRF 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](#)
- [DS881, GW1NSER 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](#)
- [DS961, GW2ANR 系列 FPGA 产品数据手册](#)
- [DS976, GW2AN-55 器件数据手册](#)
- [DS971, GW2AN-18X & 9X 器件数据手册](#)
- [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
AHB	Advanced High Performance Bus	高级高性能总线
APB	Advanced Peripheral Bus	高级外设总线

术语、缩略语	全称	含义
FIFO	First Input First Output	先进先出
FPGA	Field Programmable Gate Array	现场可编程门阵列
SPI	Serial Peripheral Interface	串行外围设备接口

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 概述

Flash 是一种非易失闪存技术，Serial Peripheral Interface (SPI) 是串行外围设备接口。SPI Flash 就是通过串行的接口进行操作的 Flash 存储设备。由于 SPI Flash 具有接口简单，封装小等特点，目前被广泛应用于音视频设备，手机，移动外设，办公设备及工控设备等产品中。

Gowin 设计一款通用 SPI Flash Interface IP，该 IP 为用户提供一个通用的命令接口，使其与 SPI Flash 芯片进行互连，完成用户的访存需求。

表 2-1 Gowin SPI Flash Interface IP

Gowin SPI Flash Interface IP	
逻辑资源	请参见表 2-2。
交付文件	
设计文件	Verilog (encrypted)
参考设计	Verilog
TestBench	Verilog
测试设计流程	
综合软件	GowinSynthesis®
应用软件	Gowin Software (V 1.9.8.10 及以上)

注！

可登录[高云半导体网站](#)查看芯片支持信息。

2.2 主要特征

- 支持 AHB 和 APB 总线配置接口
- 支持 Normal、Dual、Quad SPI 接口
- 支持 memory-mapped 访问(只读)
- 采用可配置 SPI SCLK
- 支持可配置 TX/RX FIFO 深度

2.3 资源利用

通过 Verilog 语言实现 Gowin SPI Flash Interface IP。因使用器件的密度、速度和等级不同，其性能和资源利用情况可能不同。以高云 GW1NSR-4C 系列 FPGA 为例，Gowin SPI Flash Interface IP 其资源利用情况如表 2-2 所示。

表 2-2 占用资源

器件系列	速度等级	器件名称	资源利用	备注
GW1NSR-4C	-6	LUT	814	-
		REG	361	
		BSRAM	2	

3 功能描述

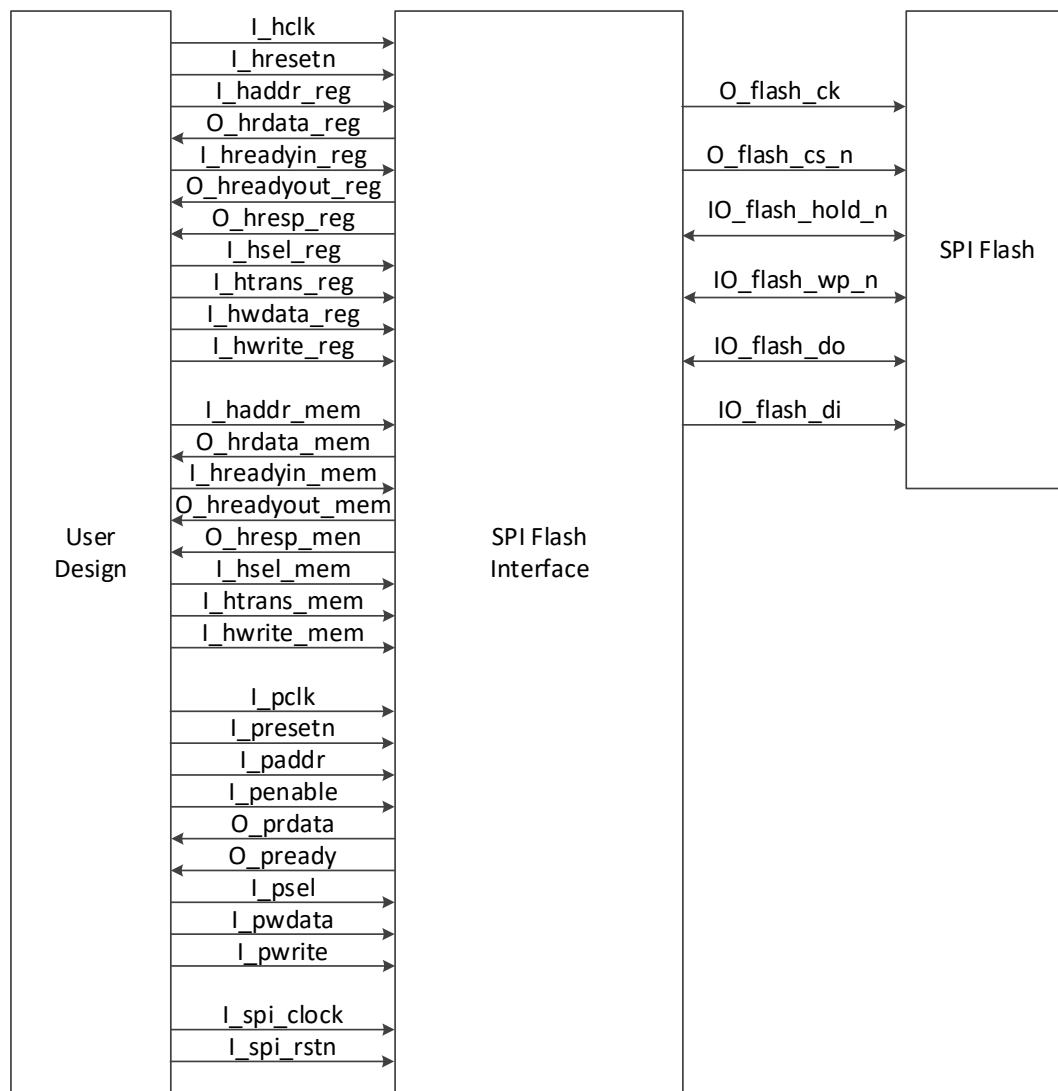
3.1 系统框图

Gowin SPI Flash Interface IP 是一个串行外围设备接口控制器 IP，作为一个主设备，与 SPI Flash 设备相连。SPI 传输格式和传输时序可通过 AHB 或 APB 总线接口进行配置。SPI 传输的数据可通过 AHB 总线接口以 memory-mapped 只读方式或寄存器读写方式操作，也可通过 APB 总线接口以寄存器读写方式操作。

对寄存器方式读写的 SPI 传输，传输格式中包含了命令段，地址段和数据段。IP 中提供了相应的寄存器列表，可以通过 AHB 或 APB 总线传递命令，地址和数据。

对 memory-mapped 只读方式的 SPI 传输，只需通过 AHB 总线传递地址，即可获取对应地址空间的数据，且只支持 AHB 总线。

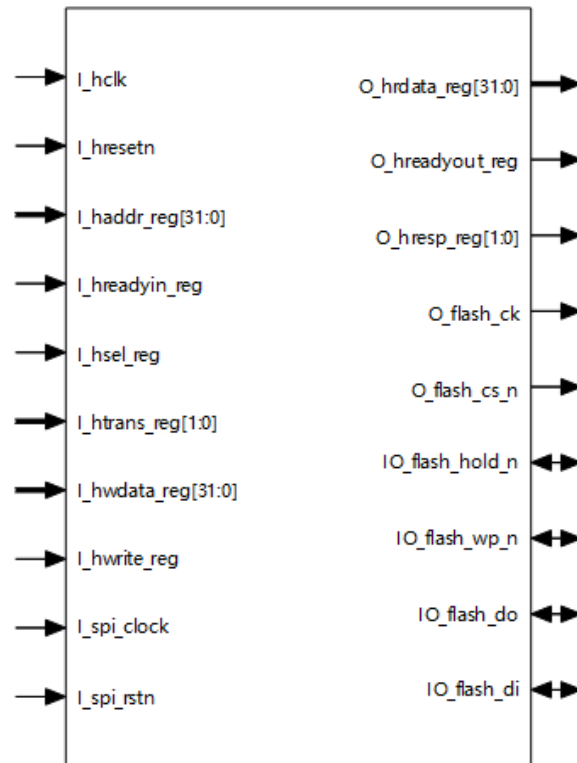
图 3-1 系统框图



3.2 端口列表

Gowin SPI Flash Interface IP 的 IO 端口如图 3-2 所示。

图 3-2 Gowin SPI Flash Interface IP IO 端口示意图



根据配置参数不同，端口会略有不同。

Gowin SPI Flash Interface IP 的 IO 端口详细描述如表 3-1 所示。

表 3-1 Gowin SPI Flash Interface IP 的端口列表

序号	信号名称	方向	描述	备注
1	I_hclk	I	AHB 总线工作时钟	所有信号输入输出方向均以 SPI Flash Interface IP 为参考。
2	I_hresetn	I	AHB 总线复位，低有效。	
3	I_haddr_reg	I	AHB reg 模式地址	
4	O_hrdata_reg	O	AHB reg 模式读数据	
5	I_hreadyin_reg	I	AHB reg 模式主设备 ready 信号	
6	O_hreadyout_reg	O	AHB reg 模式从设备 ready 信号	
7	O_hresp_reg	O	AHB reg 模式传输响应	
8	I_hsel_reg	I	AHB reg 模式从设备选择信号	
9	I_htrans_reg	I	AHB reg 模式传输类型	
10	I_hwdata_reg	I	AHB reg 模式写数据	
11	I_hwrite_reg	I	AHB reg 模式读写信号，1 表示写，0 表示读。	

序号	信号名称	方向	描述	备注
12	I_haddr_mem	I	AHB mem 模式地址 仅 memory-mapped 模式有效	
13	O_hrdata_mem	O	AHB mem 模式读数据 仅 memory-mapped 模式有效	
14	I_hreadyin_mem	I	AHB mem 模式主设备 ready 信号 仅 memory-mapped 模式有效	
15	O_hreadyout_mem	O	AHB mem 模式从设备 ready 信号 仅 memory-mapped 模式有效	
16	O_hresp_mem	O	AHB mem 模式传输响应 仅 memory-mapped 模式有效	
17	I_hsel_mem	I	AHB mem 模式从设备选择信号 仅 memory-mapped 模式有效	
18	I_htrans_mem	I	AHB mem 模式传输类型 仅 memory-mapped 模式有效	
19	I_hwrite_mem	I	AHB mem 模式读写信号, 0 表示读, 无写操作 仅 memory-mapped 模式有效	
20	I_pclk	I	APB 总线工作时钟	
21	I_presetn	I	APB 总线复位, 低有效。	
22	I_paddr	I	APB reg 模式地址	
23	I_penable	I	APB reg 模式从设备使能信号	
24	O_prdata	O	APB reg 模式读数据	
25	O_pready	O	APB reg 模式从设备 ready 信号	
26	I_psel	I	APB reg 模式从设备选择信号	
27	I_pwdata	I	APB reg 模式写数据	
28	I_pwrite	I	APB reg 模式读写信号, 1 表示写, 0 表示读。	
29	I_spi_clock	I	SPI Flash 参考时钟	
30	I_spi_rstn	I	SPI Flash 复位, 低有效。	
31	O_flash_ck	O	SPI Flash 时钟信号	
32	O_flash_cs_n	O	SPI Flash 片选信号	
33	IO_flash_hold_n	IO	SPI Flash 保持信号	
34	IO_flash_wp_n	IO	SPI Flash 写保护信号	
35	IO_flash_do	I/O	SPI Flash 数据信号 MISO	
36	IO_flash_di	O	SPI Flash 数据信号 MOSI	

3.3 参数配置

表 3-2 Gowin SPI Flash Interface IP 参数

序号	参数名称	允许范围	默认值	描述
1	Bus Type	AHB/APB	AHB	AHB 表示支持 AHB 总线 APB 表示支持 APB 总线
2	Memory-mapped Read Support	Yes/No	No	表示是否支持 memory-mapped 读取模式，数据只读。
3	SPI Mode	Normal/Dual/Quad	Normal	<ul style="list-style-type: none"> ● Normal: 支持标准 SPI ● Dual: 支持 Dual SPI ● Quad: 支持 Quad SPI
4	SPI Clock Divider	0~128	0	表示 SCLK 频率与源时钟频率比率 当 SPI Clock Divider 值为 N, N 为不同值时对应关系如下: <ul style="list-style-type: none"> ● 0: $F_{sclk} = F_{source_clk}$ ● 1~128: $F_{sclk} = F_{source_clk} / (N * 2)$
5	TX FIFO Depth	2/4/8/16/32/64/128	4	RX FIFO 深度
6	RX FIFO Depth	2/4/8/16/32/64/128	4	TX FIFO 深度

3.4 寄存器描述

表 3-3 寄存器列表

地址偏移	寄存器名	描述
0x00~0x0C	-	保留
0x10	TransFmt	SPI 传输格式寄存器
0x20	TransCtrl	SPI 传输控制寄存器
0x24	Cmd	SPI 命令寄存器
0x28	Addr	SPI 地址寄存器
0x2C	Data	SPI 数据寄存器
0x30	Ctrl	SPI 控制寄存器
0x34	Status	SPI 状态寄存器
0x38	IntrEn	SPI 中断使能寄存器
0x3C	IntrSt	SPI 中断状态寄存器
0x40	Timing	SPI 接口时序寄存器

地址偏移	寄存器名	描述
0x44~0x78	-	保留
0x7C	Config	配置寄存器

注!

寄存器读写类型定义:

- RO: 只读
- RW: 可读可写
- W1C: 可读, 写 1 清零。

3.4.1 SPI 传输格式寄存器(0x10)

表 3-4 SPI 传输格式寄存器

名称	比特位	类型	描述	默认值
Reserved	31:18	-	-	0x0
AddrLen	17:16	RW	地址长度(单位 byte) 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	0x2
Reserved	15:13	-	-	0x0
DataLen	12:8	RW	数据长度(单位 bit) 实际数据长度为(DataLen+1)bits	0x07
DataMerge	7	RW	数据合并模式使能, 只在 DataLen = 0x07 时有效 ● 0x1: 4 bytes 传输 ● 0x0: 只传输最低字节	0x1
Reserved	6:4	-	-	0x0
LSB	3	RW	数据传输顺序 ● 0x0: 最高有效位优先 ● 0x1: 最低有效位优先	0x0
Reserved	2:0	-	-	0x0

注!

寄存器操作开始前, 必须首先设置 SPI 传输格式寄存器的值, 例如: 3 字节地址, 数据长度为 8, 设置 SPI 传输格式寄存器值为 0x00020780。

3.4.2 SPI 传输控制寄存器(0x20)

表 3-5 SPI 传输控制寄存器

名称	比特位	类型	描述	默认值
Reserved	31	-	-	0x0
CmdEn	30	RW	SPI 命令段使能 ● 0x0: 禁止命令段 ● 0x1: 使能命令段	0x0
AddrEn	29	RW	SPI 地址段使能	0x0

名称	比特位	类型	描述	默认值
			<ul style="list-style-type: none"> ● 0x0: 禁止地址段 ● 0x1: 使能地址段 	
AddrFmt	28	RW	SPI 地址段格式 <ul style="list-style-type: none"> ● 0x0: Single 模式 ● 0x1: 格式同数据段格式 	0x0
TransMode	27:24	RW	传输模式 <ul style="list-style-type: none"> ● 0x0: 同时写和读 ● 0x1: 只写 ● 0x2: 只读 ● 0x3: 写/读 ● 0x4: 读/写 ● 0x5: 写/Dummy/读 ● 0x6: 读/Dummy/写 ● 0x7: 无数据(必须使能 CmdEn 或 AddrEn) ● 0x8: Dummy/写 ● 0x9: Dummy/读 ● 0xa~0xf: 保留 	0x0
DualQuad	23:22	RW	SPI 数据段格式 <ul style="list-style-type: none"> ● 0x0: Single 模式 ● 0x1: Dual I/O 模式 ● 0x2: Quad I/O 模式 ● 0x3: 保留 	0x0
TokenEn	21	RW	Token 字节传输使能 SPI 读操作时在地址段后附加 1 个字节, 值由 TokenValue 选择。 <ul style="list-style-type: none"> ● 0x0: 禁止附加 1 字节 ● 0x1: 使能附加 1 字节 	0x0
WrTranCnt	20:12	RW	写入数据单元数量。 <ul style="list-style-type: none"> ● WrTranCnt 指示从数据寄存器传输到 SPI 总线的数据单元数。实际数量是(WrTranCnt+1)。 ● WrTranCnt 只在 TransMode 为 0,1,3,4,5,6 或 8 时起效。 ● 对 TransMode 0, WrTranCnt 必须等于 RdTranCnt。 	0x0
TokenValue	11	RW	Token 字节值 <ul style="list-style-type: none"> ● 0x0: 值=0x00 ● 0x1: 值=0x69 	0x0
DummyCnt	10:9	RW	Dummy 数据数量。 <ul style="list-style-type: none"> ● 实际数量是(DummyCnt+1)。 	0x0

名称	比特位	类型	描述	默认值
			<ul style="list-style-type: none"> ● SPI 接口端 Dummy 周期数是 $(\text{DummyCnt}+1) \times ((\text{DataLen}+1)/\text{SPI IO width})$ ● 在 Dummy 期数据脚为高阻。 ● DummyCnt 只在 TransMode 为 5,6,8,9 时有效。 ● 普通传输格式 Dummy 周期设置如表 3-6 所示。 	
RdTranCnt	8:0	RW	读取数据单元数量。 <ul style="list-style-type: none"> ● RdTranCnt 指示从 SPI 总线接收并存储到数据寄存器的数据单元数。实际数量是 $(\text{RdTranCnt}+1)$。 ● RdTranCnt 只在 TransMode 为 0,2,3,4,5,6 或 9 时起效。 ● 对 TransMode 0, WrTranCnt 必须等于 RdTranCnt。 	0x0

表 3-6 普通传输格式 Dummy 周期设置

传输格式	DummyCnt+1	DataLen+1	DualQuad	#Dummy Cycles on the SPI Interface
Dummy 周期	1	8	Single	8
	1	8	Dual	4
	1	8	Quad	2
	2	8	Quad	4
	3	8	Quad	6
	1	32	Quad	8

3.4.3 SPI 命令寄存器(0x24)

表 3-7 SPI 命令寄存器

名称	比特位	类型	描述	默认值
Reserved	31:8	-	-	0x0
CMD	7:0	RW	SPI 命令	0x0

3.4.4 SPI 地址寄存器(0x28)

表 3-8 SPI 地址寄存器

名称	比特位	类型	描述	默认值
ADDR	31:0	RW	SPI 地址	0x0

3.4.5 SPI 数据寄存器(0x2C)

表 3-9 SPI 数据寄存器

名称	比特位	类型	描述	默认值
DATA	31:0	RW	<p>SPI 发送或接收数据</p> <p>对写操作，数据写入到 TX FIFO。低字节先传送。如果 TX FIFO 满，且状态寄存器的 SPIActive 位为 1，则 hready 信号拉低来插入等待状态。</p> <p>对读操作，从 RX FIFO 读取数据。低字节先传送。如果 RX FIFO 空，且状态寄存器的 SPIActive 位为 1，则 hready 信号拉低来插入等待状态。</p> <p>当 TX FIFO 为空，SPI 传输将保持住直到更多数据写入 TX FIFO；当 RX FIFO 为满，SPI 传输将保持住直到 RX FIFO 有空间可接收数据。</p> <p>在写操作中，如果超过 WrTranCnt 的数据被写入 TX FIFO 中，多余的数据会进入到下次的传输，或通过复位 TX FIFO 清除。</p>	0x0

3.4.6 SPI 控制寄存器(0x30)

表 3-10 SPI 控制寄存器

名称	比特位	类型	描述	默认值
Reserved	31:3	-	-	0x0
TXFIFORST	2	RW	<p>发送 FIFO 复位</p> <p>写 1 复位，在复位操作完成后自动清零。</p>	0x0
RXFIFORST	1	RW	<p>接收 FIFO 复位</p> <p>写 1 复位，在复位操作完成后自动清零。</p>	0x0
SPIRST	0	RW	<p>SPI 复位</p> <p>写 1 复位，在复位操作完成后自动清零。</p>	0x0

3.4.7 SPI 状态寄存器(0x34)

表 3-11 SPI 状态寄存器

名称	比特位	类型	描述	默认值
Reserved	31:30	-	-	0x0
TXNUM[7:6]	29:28	RO	发送 FIFO 中有效数据数量[7:6]	0x0
Reserved	27:26	-	-	0x0
RXNUM[7:6]	25:24	RO	接收 FIFO 中有效数据数量[7:6]	0x0
TXFULL	23	RO	发送 FIFO 满标志	0x0

名称	比特位	类型	描述	默认值
TXEMPTY	22	RO	发送 FIFO 空标志	0x1
TXNUM[5:0]	21:16	RO	发送 FIFO 中有效数据数量[5:0]	0x0
RXFULL	15	RO	接收 FIFO 满标志	0x0
RXEMPTY	14	RO	接收 FIFO 空标志	0x1
RXNUM[5:0]	13:8	RO	接收 FIFO 中有效数据数量[5:0]	0x0
Reserved	7:1	-	-	0x0
SPIActive	0	RO	SPI 寄存器正在操作标志 当 SPI 命令寄存器被写入后， SPIActive 变为 1，在传输完成后， SPIActive 变为 0。	0x0

3.4.8 SPI 中断使能寄存器(0x38)

表 3-12 SPI 中断使能寄存器

名称	比特位	类型	描述	默认值
Reserved	31:5	-	-	0x0
EndIntEn	4	RW	SPI 传输结束中断使能 控制当 SPI 传输结束时是否触发中断。	0x0
Reserved	3:0	-	-	0x0

3.4.9 SPI 中断状态寄存器(0x3C)

表 3-13 SPI 中断状态寄存器

名称	比特位	类型	描述	默认值
Reserved	31:5	-	-	0x0
EndInt	4	W1C	SPI 传输结束中断 当 SPI 传输结束中断触发时该 bit 位设为 1。	0x0
Reserved	3:0	-	-	0x0

3.4.10 SPI 接口时序寄存器(0x40)

表 3-14 SPI 接口时序寄存器

名称	比特位	类型	描述	默认值
Reserved	31:12	-	-	0x0
Reserved	11:8	-	-	0x2
SCLK_DIV	7:0	RW	SPI 接口时钟与源时钟频率比率，取决于配置 <ul style="list-style-type: none"> 当 SCLK_DIV 值在范围 0~127 时，SCLK 频率=源时钟频率 / ((SCLK_DIV+1)*2) 当 SCLK_DIV 值为 255 时，SCLK 频率等于源时钟 spi_clock 频率。 	0xff

3.4.11 配置寄存器(0x7C)

表 3-15 配置寄存器

名称	比特位	类型	描述	默认值
Reserved	31:8	-	-	0x0
TxFIFOSize	7:4	RO	TX FIFO 深度，取决于配置。 <ul style="list-style-type: none"> ● 0x0: 2 words ● 0x1: 4 words ● 0x2: 8 words ● 0x3: 16 words ● 0x4: 32 words ● 0x5: 64 words ● 0x6: 128 words 	0x1
RXFIFOSize	3:0	RO	RX FIFO 深度，取决于配置。 <ul style="list-style-type: none"> ● 0x0: 2 words ● 0x1: 4 words ● 0x2: 8 words ● 0x3: 16 words ● 0x4: 32 words ● 0x5: 64 words ● 0x6: 128 words 	0x1

3.5 硬件配置选项

3.5.1 TX FIFO 深度

通过 IP 内核生成器来确定 TX FIFO 的深度，对应配置寄存器(0x7C)中 TxFIFOSize，深度可设为 2,4,8,16,32,64,128。默认深度值为 4。

3.5.2 RX FIFO 深度

通过 IP 内核生成器来确定 RX FIFO 的深度，对应配置寄存器(0x7C)中 RXFIFOSize，深度可设为 2,4,8,16,32,64,128。默认深度值为 4。

3.5.3 Memory-Mapped 读取

IP 提供了 memory-mapped 读取的接口。通过 IP 内核生成器确定是否支持该接口。

3.5.4 SPI 接口时钟参数

通过 IP 内核生成器来确定 SPI 接口时钟频率，对应 SPI 接口时序寄存器(0x40)中的 SCLK_DIV。例如：

当 SPI Clock Divider 为 0，SCLK_DIV 值为 255。

当 SPI Clock Divider 为 1~128，SCLK_DIV 值为 0~127。

3.6 读写操作流程

本文以 SPI Flash W25Q64DW 为例，说明 SPI Flash Interface IP 的数据读写操作步骤，SPI Flash 部分命令列表如下：

表 3-16 SPI Flash 部分命令列表

命令	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
Normal SPI						
Write Enable	06H	-	-	-	-	-
Write Disable	04H	-	-	-	-	-
Read Status Register-1	05H	(S7-S0)	-	-	-	-
Read Status Register-2	35H	(S15-S8)	-	-	-	-
Write Status Register	01H	(S7-S0)	(S15-S8)	-	-	-
Page Program	02H	A23-A16	A15-A8	A7-A0	D7-D0	D7-D0
Sector Erase(4KB)	20H	A23-A16	A15-A8	A7-A0	-	-
Block Erase(32KB)	52H	A23-A16	A15-A8	A7-A0	-	-
Block Erase(64KB)	D8H	A23-A16	A15-A8	A7-A0	-	-
Chip Erase	C7/60H	-	-	-	-	-
Read Data	03H	A23-A16	A15-A8	A7-A0	(D7-D0)	
Read Manufacturer/ Device ID	90H	dummy	dummy	00H	(MF7-MF0)	(ID7-ID0)
Read JEDEC ID	9FH	(MF7-MF0)	(ID15-ID8)	(ID7-ID0)	-	-
Read Unique ID	4BH	dummy	dummy	dummy	dummy	(UID63- UID0)
Dual SPI						
Fast Read Dual Output	3BH	A23-A16	A15-A8	A7-A0	dummy	(D7- D0,...)
Fast Read Dual I/O	BBH	A23-A8	A7-A0,M7- M0	(D7- D0,...)	-	-
Read Manufacturer/Device ID by Dual I/O	92H	A23-A8	A7-A0,M7- M0	(MF7- MF0,ID7- ID0)	-	-
Quad SPI						
Quad Page Program	32H	A23-A16	A15-A8	A7-A0	D7-D0	D7-D0
Fast Read Quad Output	6BH	A23-A16	A15-A8	A7-A0	dummy	(D7- D0,...)
Fast Read Quad I/O	EBH	A23-A0,M7- M0	(xxxx,D7- D0)	(D7- D0,...)	-	-
Read Manufacturer/Device ID by Quad I/O	94H	A23-A0,M7- M0	xxxx,(MF7- MF0,ID7- ID0)	(MF7- MF0,ID7- ID0)	-	-

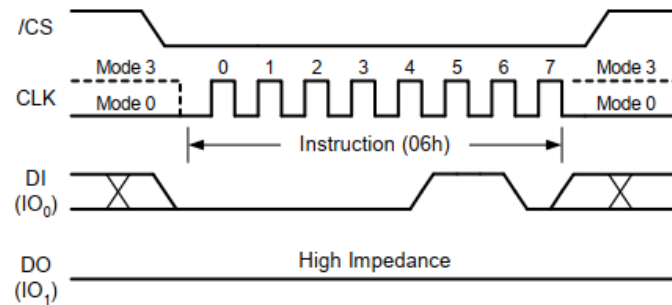
3.6.1 Write Enable (06H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - a) CmdEn = 1
 - b) TransMode = 0x7 (无数据)

2. 设置 SPI 命令寄存器(0x24)值为 0x00000006
CMD = 0x06

图 3-3 Write Enable 时序图

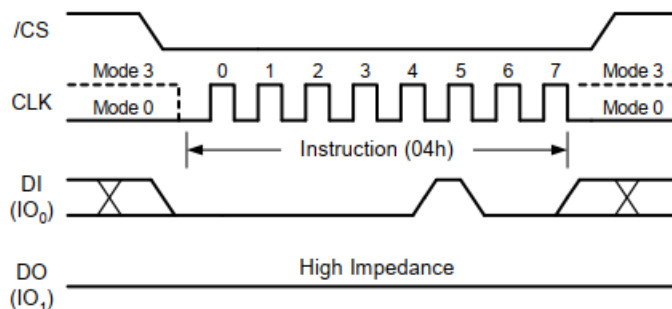


3.6.2 Write Disable (04H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - a) CmdEn = 1
 - b) TransMode = 0x7 (无数据)
2. 设置 SPI 命令寄存器(0x24)值为 0x00000004
CMD = 0x04

图 3-4 Write Disable 时序图

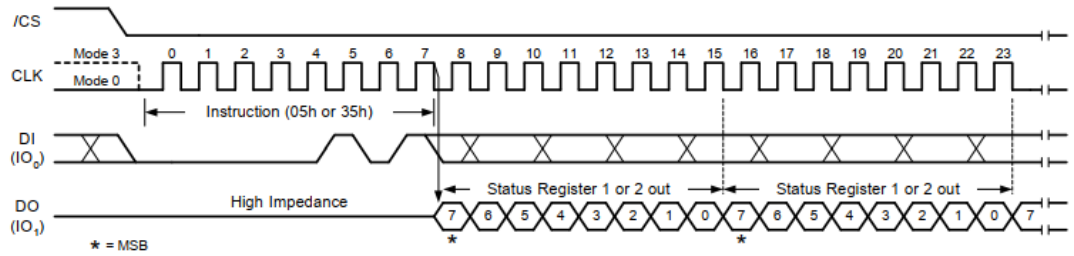


3.6.3 Read Status Register (05H/35H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x42000000
 - a) CmdEn = 1
 - b) TransMode = 0x2 (只读)
2. 设置 SPI 命令寄存器(0x24)值为 0x00000005
CMD = 0x05
3. 读取 SPI 数据寄存器(0x2C)

图 3-5 Read Status Register 时序图

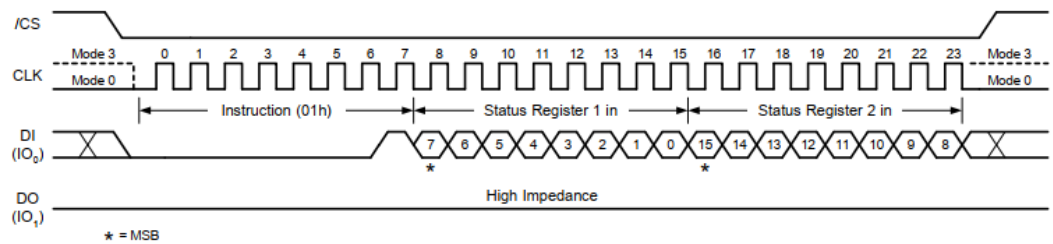


3.6.4 Write Status Register (01H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x41000000
 - a) CmdEn = 1
 - b) TransMode = 0x1 (只写)
2. 设置 SPI 数据寄存器(0x2C)值为 0x00000200
DATA = 0x00000200 (关闭 Protect Bit, QE bit 保持 1)
3. 设置 SPI 命令寄存器(0x24)值为 0x00000001
CMD = 0x01

图 3-6 Write Status Register 时序图



3.6.5 Read Data (03H)操作过程

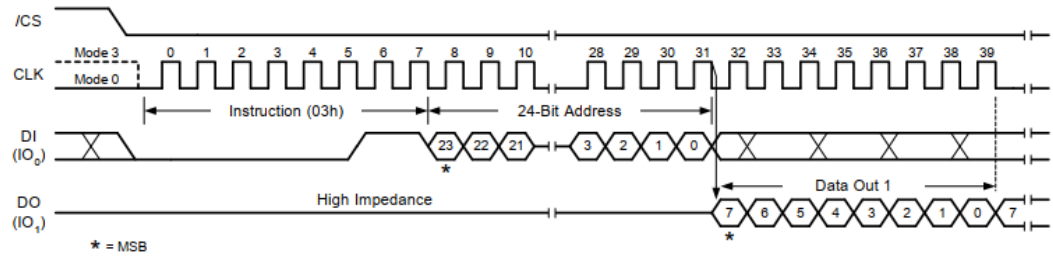
举例连续读取 16 个字节具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x6200000f
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x2 (只读)
 - d) RdTranCnt = 15 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x00000003
CMD = 0x03

5. 读取 SPI 数据寄存器(0x2C)

每次读取最多 4 个字节，超过 4 个字节，分多次读取此寄存器。

图 3-7 Read Data 时序图



3.6.6 Page Program (02H)操作过程

举例连续写入 16 个字节具体工作过程，主要分为两大步骤，第一擦除数据，第二写入数据，具体命令如下：

1. 擦除数据

- a) 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - i. CmdEn = 1
 - ii. TransMode = 0x7 (无数据)
- b) 设置 SPI 命令寄存器(0x24)值为 0x00000006
CMD = 0x06
- c) 设置 SPI 传输控制寄存器(0x20)值为 0x67000000
 - i. CmdEn = 1
 - ii. AddrEn = 1
 - iii. TransMode = 0x7 (无数据)
- d) 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
- e) 设置 SPI 命令寄存器(0x24)值为 0x00000020
CMD = 0x20 (扇区擦除)

2. 写入数据

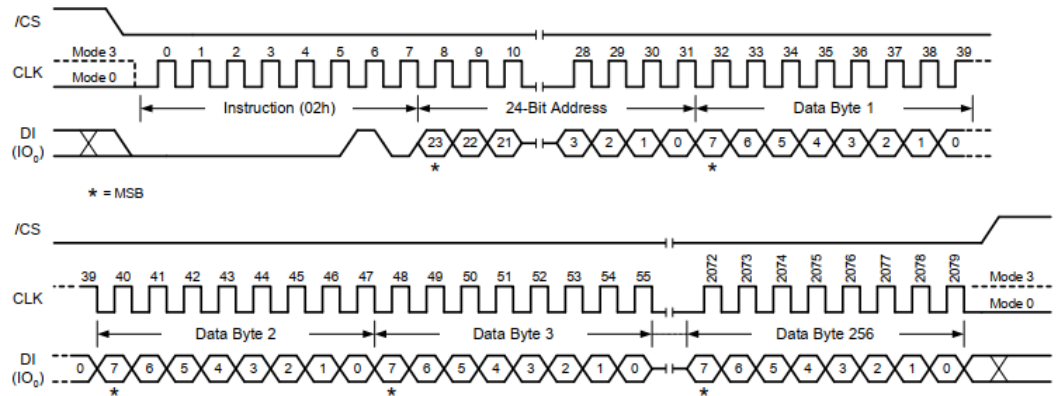
- a) 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - i. CmdEn = 1
 - ii. TransMode = 0x7 (无数据)
- b) 设置 SPI 命令寄存器(0x24)值为 0x00000006
CMD = 0x06
- c) 设置 SPI 传输控制寄存器(0x20)值为 0x6100f000
 - i. CmdEn = 1
 - ii. AddrEn = 1

- iii. TransMode = 0x1 (只写)
- iv. WrTranCnt = 15 (传输总数-1)
- d) 设置 SPI 控制寄存器(0x30)值为 0x00000004
TXFIFORST = 1
- e) 设置 SPI 中断使能寄存器(0x38)值为 0x00000010
EndIntEn = 1
- f) 设置 SPI 数据寄存器(0x2C)值为 0x33221100
DATA = 0x33221100 (1~4 字节)
- g) 设置 SPI 数据寄存器(0x2C)值为 0x77665544
DATA = 0x77665544 (5~8 字节)
- h) 设置 SPI 数据寄存器(0x2C)值为 0x33221100
DATA = 0xbbaa9988 (9~12 字节)
- i) 设置 SPI 数据寄存器(0x2C)值为 0x77665544
DATA = 0xffeeddcc (13~16 字节)
- j) 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
- k) 设置 SPI 命令寄存器(0x24)值为 0x00000002
CMD = 0x02
- l) 读取 SPI 中断状态寄存器(0x3C)
看 EndInt 是否为 1, 如果为 1 表示传输结束
- m) 设置 SPI 中断状态寄存器(0x3C)值为 0x00000010
EndInt= 1 (写 1 清零)

注!

- 擦除命令是将 Flash 的相应区域数据设为 1, 有 4 个擦除命令, 扇区擦除, 32KB 块擦除, 64KB 块擦除, 片擦除。如果被擦除区域有其他有效数据, 操作者写入时需先读取被擦除区域的其他数据, 及时回写。
- 写入数据量不能超过 TX FIFO 深度。

图 3-8 Page Program 时序图



3.6.7 Sector Erase (20H)操作过程

具体工作过程:

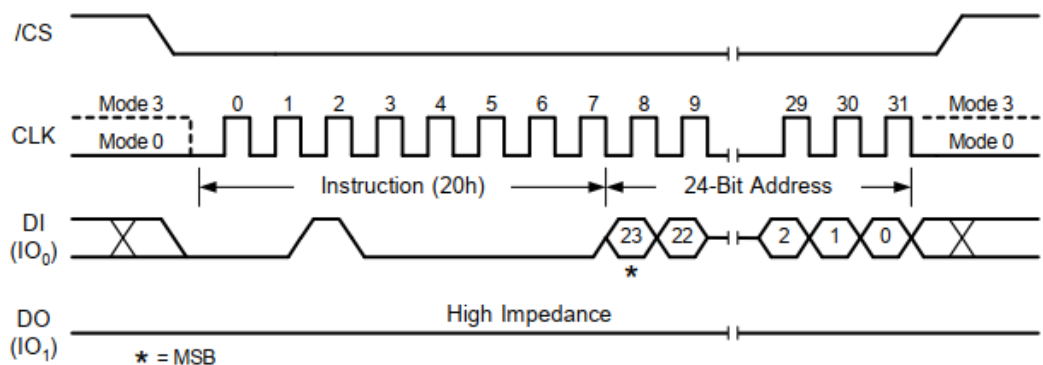
1. 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - a) CmdEn = 1
 - b) TransMode = 0x7 (无数据)
2. 设置 SPI 命令寄存器(0x24)值为 0x00000006

CMD = 0x06
3. 设置 SPI 传输控制寄存器(0x20)值为 0x67000000
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x7 (无数据)
4. 设置 SPI 地址寄存器(0x28)值为 0x00000000

ADDR = 0x00000000
5. 设置 SPI 命令寄存器(0x24)值为 0x00000020

CMD = 0x20 (扇区擦除)

图 3-9 Sector Erase 时序图



3.6.8 32 KB Block Erase (52H)操作过程

具体工作过程:

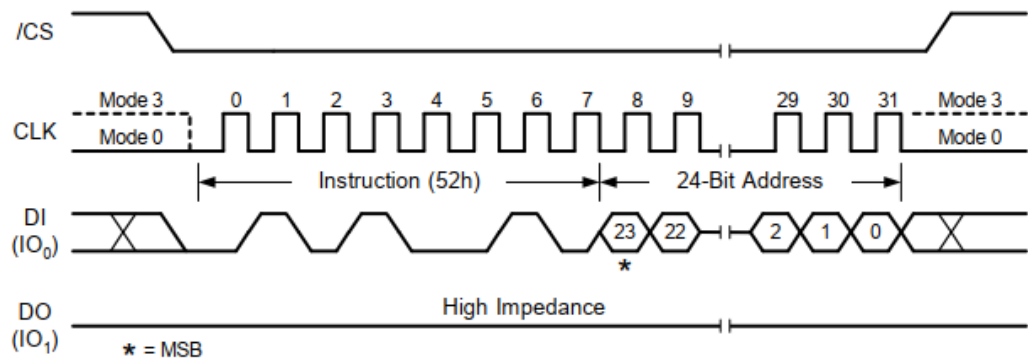
1. 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - a) CmdEn = 1
 - b) TransMode = 0x7 (无数据)
2. 设置 SPI 命令寄存器(0x24)值为 0x00000006

CMD = 0x06
3. 设置 SPI 传输控制寄存器(0x20)值为 0x67000000
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x7 (无数据)
4. 设置 SPI 地址寄存器(0x28)值为 0x00000000

ADDR = 0x00000000
5. 设置 SPI 命令寄存器(0x24)值为 0x00000052

CMD = 0x52 (32KB 块擦除)

图 3-10 32KB Block Erase 时序图



3.6.9 64 KB Block Erase (D8H)操作过程

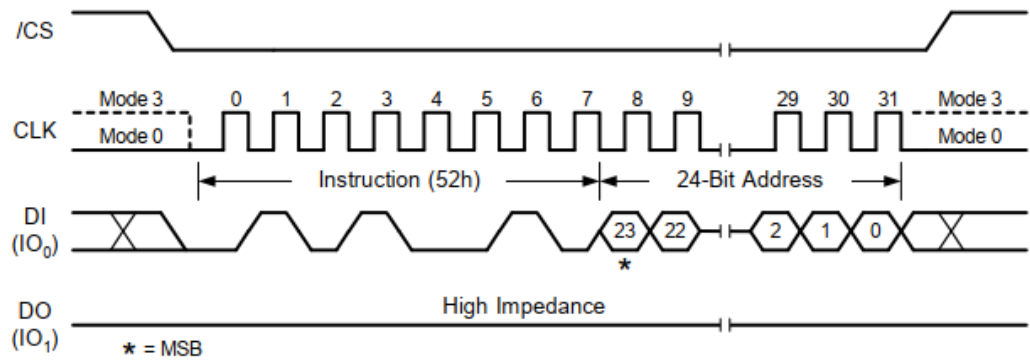
具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - a) CmdEn = 1
 - b) TransMode = 0x7 (无数据)
2. 设置 SPI 命令寄存器(0x24)值为 0x00000006

CMD = 0x06
3. 设置 SPI 传输控制寄存器(0x20)值为 0x67000000
 - a) CmdEn = 1
 - b) AddrEn = 1

- c) TransMode = 0x7 (无数据)
- 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
 - 设置 SPI 命令寄存器(0x24)值为 0x000000D8
CMD = 0xD8 (64KB 块擦除)

图 3-11 64KB Block Erase 时序图

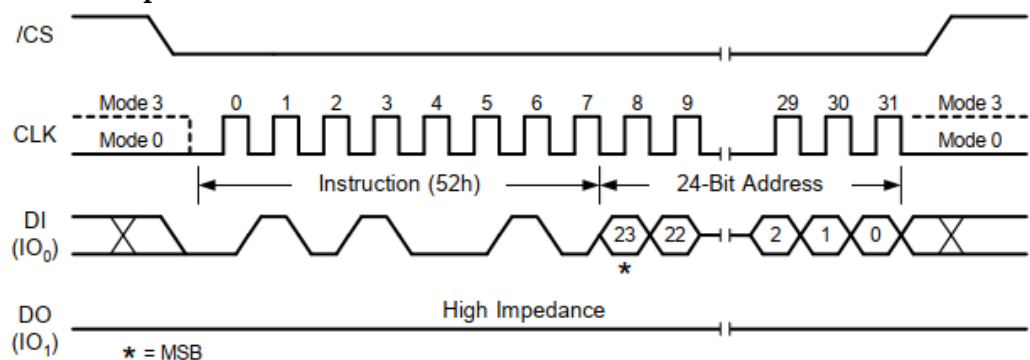


3.6.10 Chip Erase (60/C7H)操作过程

具体工作过程:

- 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - CmdEn = 1
 - TransMode = 0x7 (无数据)
- 设置 SPI 命令寄存器(0x24)值为 0x00000006
CMD = 0x06
- 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - CmdEn = 1
 - TransMode = 0x7 (无数据)
- 设置 SPI 命令寄存器(0x24)值为 0x00000060
CMD = 0x60 (片擦除)

图 3-12 Chip Erase 时序图

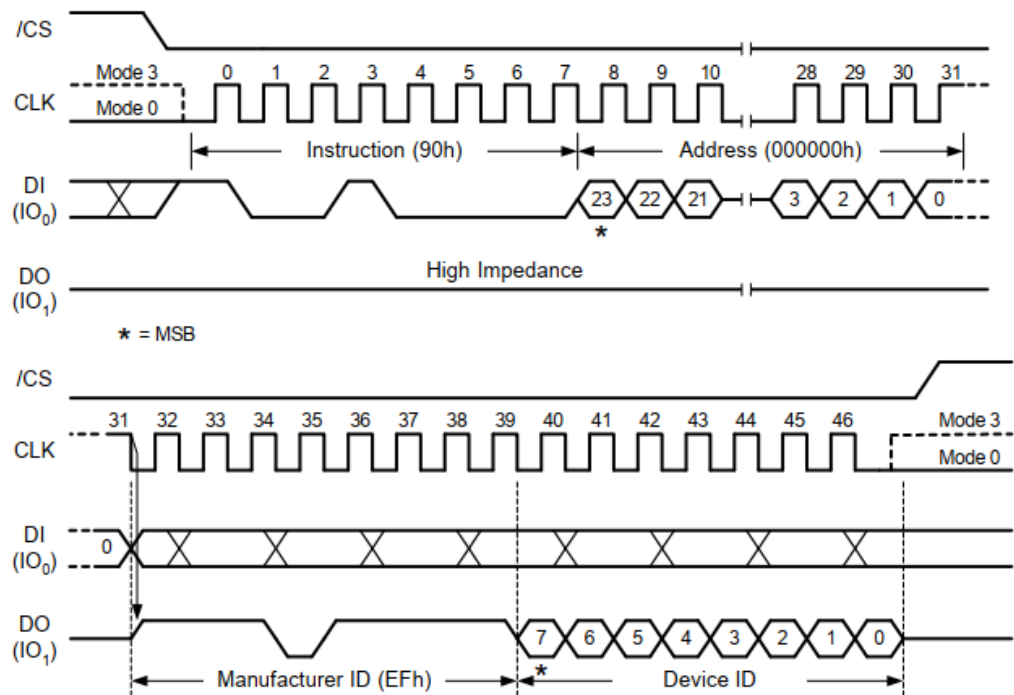


3.6.11 Read Manufacturer/Device ID (90H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x62000001
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x2 (只读)
 - d) RdTranCnt = 1 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x00000090
CMD = 0x90
5. 读取 SPI 数据寄存器(0x2C)

图 3-13 Read Manufacture ID/Device ID 时序图



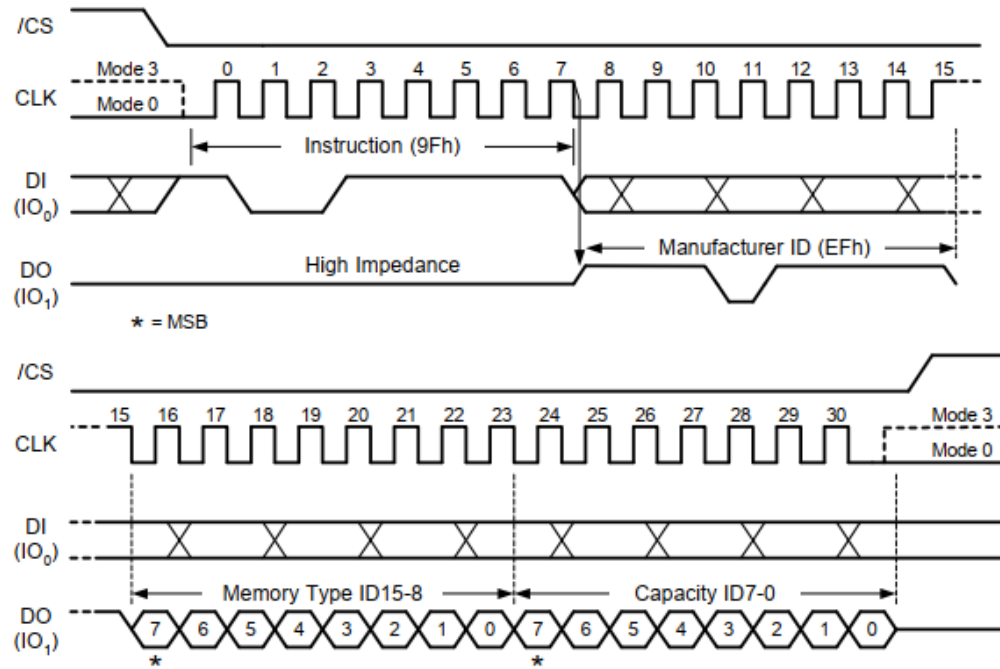
3.6.12 Read JEDEC ID (9FH)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x42000002
 - a) CmdEn = 1
 - b) TransMode = 0x2 (只读)

- c) RdTranCnt = 2 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 命令寄存器(0x24)值为 0x0000009F
CMD = 0x9F
4. 读取 SPI 数据寄存器(0x2C)

图 3-14 Read JEDEC ID 时序图



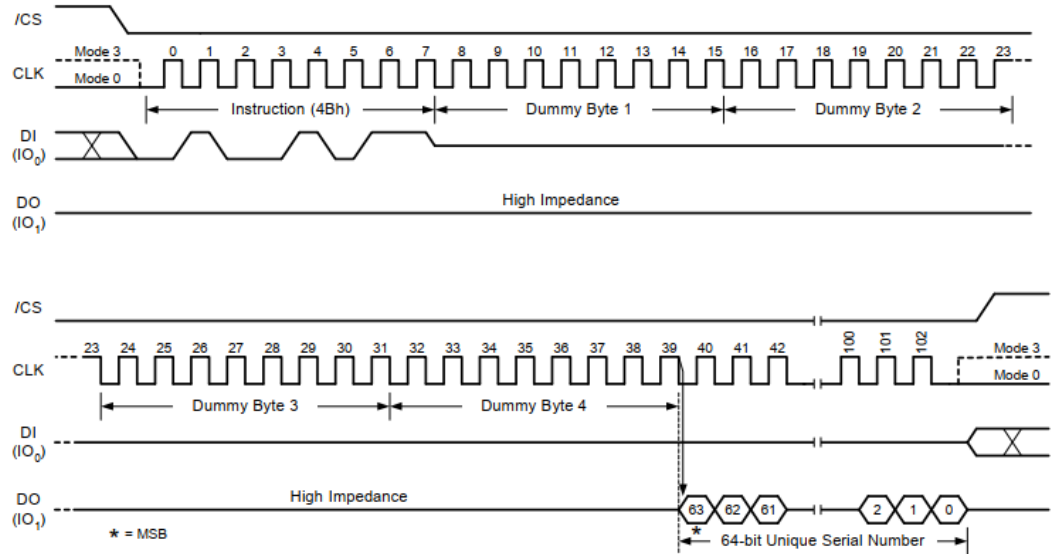
3.6.13 Read Unique ID (4BH)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x69000007
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x9 (Dummy, 读)
 - d) RdTranCnt = 7 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x0000004B
CMD = 0x4B
5. 读取 SPI 数据寄存器(0x2C)

因为 Unique ID 是 64 bits，所以需读 2 次 SPI 数据寄存器。

图 3-15 Read Unique ID 时序图

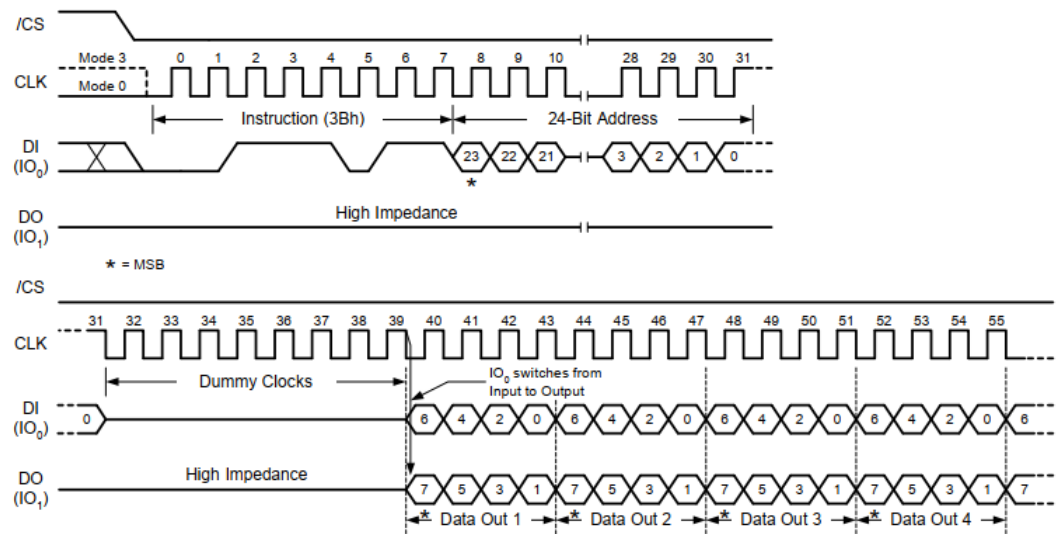


3.6.14 Fast Read Dual Output (3BH)操作过程

举例连续读取 4 个字节具体工作过程：

1. 设置 SPI 传输控制寄存器(0x20)值为 0x69400003
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x9 (Dummy, 读)
 - d) DualQuad = 0x1(Dual I/O 模式)
 - e) RdTranCnt = 3 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x0000003B
CMD = 0x3B
5. 读取 SPI 数据寄存器(0x2C)

图 3-16 Fast Read Dual Output 时序图

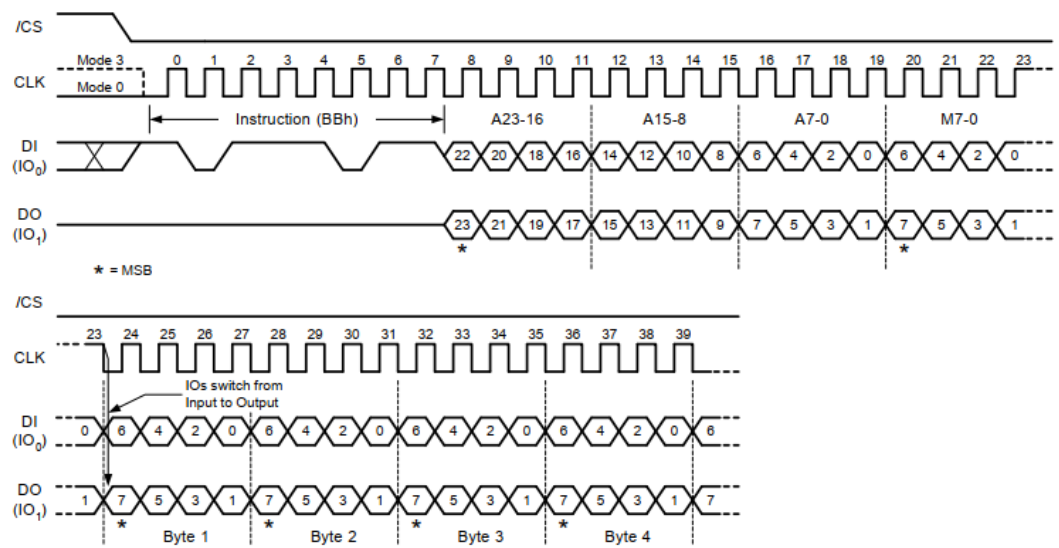


3.6.15 Fast Read Dual I/O (BBH)操作过程

举例连续读取 4 个字节具体工作过程：

1. 设置 SPI 传输控制寄存器(0x20)值为 0x79400003
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) AddrFmt = 1
 - d) TransMode = 0x9 (Dummy, 读)
 - e) DualQuad = 0x1(Dual I/O 模式)
 - f) RdTranCnt = 3 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x000000BB
CMD = 0xBB
5. 读取 SPI 数据寄存器(0x2C)

图 3-17 Fast Read Dual I/O 时序图

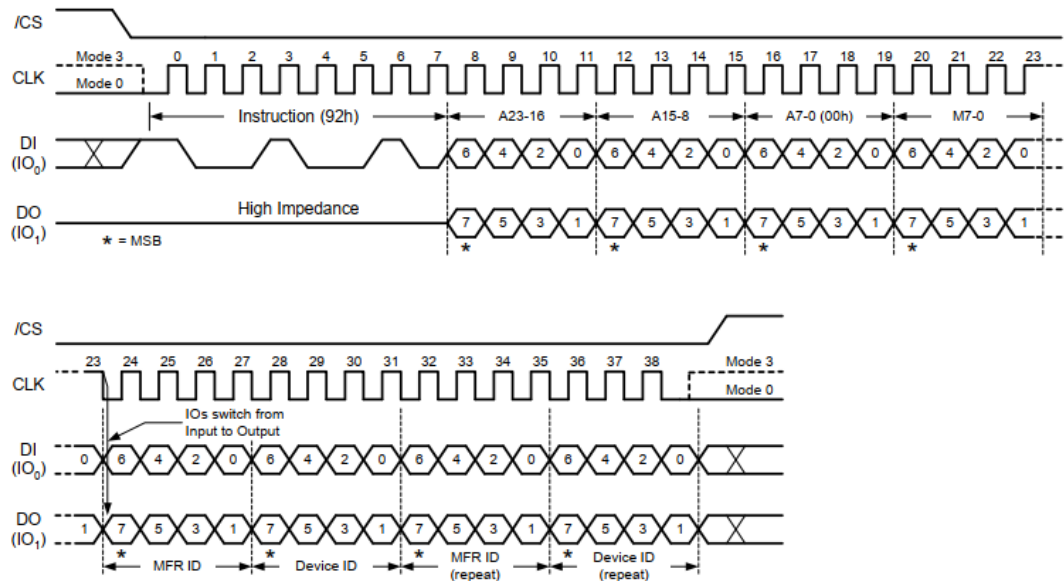


3.6.16 Read Manufacturer/Device ID Dual I/O (92H)操作过程

具体工作过程:

1. 设置 SPI 传输控制寄存器(0x20)值为 0x72600001
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) AddrFmt = 1
 - d) TransMode = 0x2 (只读)
 - e) DualQuad = 0x1(Dual I/O 模式)
 - f) TokenEn = 1(Token 字节使能)
 - g) RdTranCnt = 1 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x00000092
CMD = 0x92
5. 读取 SPI 数据寄存器(0x2C)

图 3-18 Read Manufacturer/Device ID Dual I/O 时序图



3.6.17 Quad Page Program (32H)操作过程

举例连续写入 4 个字节具体工作过程，主要分为两大步骤，第一擦除数据，第二写入数据，具体命令如下：

1. 擦除数据

- a) 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - i. CmdEn = 1
 - ii. TransMode = 0x7 (无数据)
- b) 设置 SPI 命令寄存器(0x24)值为 0x00000006
CMD = 0x06
- c) 设置 SPI 传输控制寄存器(0x20)值为 0x67000000
 - i. CmdEn = 1
 - ii. AddrEn = 1
 - iii. TransMode = 0x7 (无数据)
- d) 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
- e) 设置 SPI 命令寄存器(0x24)值为 0x00000020
CMD = 0x20 (扇区擦除)

2. 写入数据

- a) 设置 SPI 传输控制寄存器(0x20)值为 0x47000000
 - i. CmdEn = 1
 - ii. TransMode = 0x7 (无数据)
- b) 设置 SPI 命令寄存器(0x24)值为 0x00000006

CMD = 0x06

c) 设置 SPI 传输控制寄存器(0x20)值为 0x61803000

- i. CmdEn = 1
- ii. AddrEn = 1
- iii. TransMode = 0x1 (只写)
- iv. DualQuad = 0x2(Quad I/O 模式)
- v. WrTranCnt = 3 (传输总数-1)

d) 设置 SPI 数据寄存器(0x2C)值为 0x33221100

DATA = 0x33221100 (1~4 字节)

e) 设置 SPI 地址寄存器(0x28)值为 0x00000000

ADDR = 0x00000000

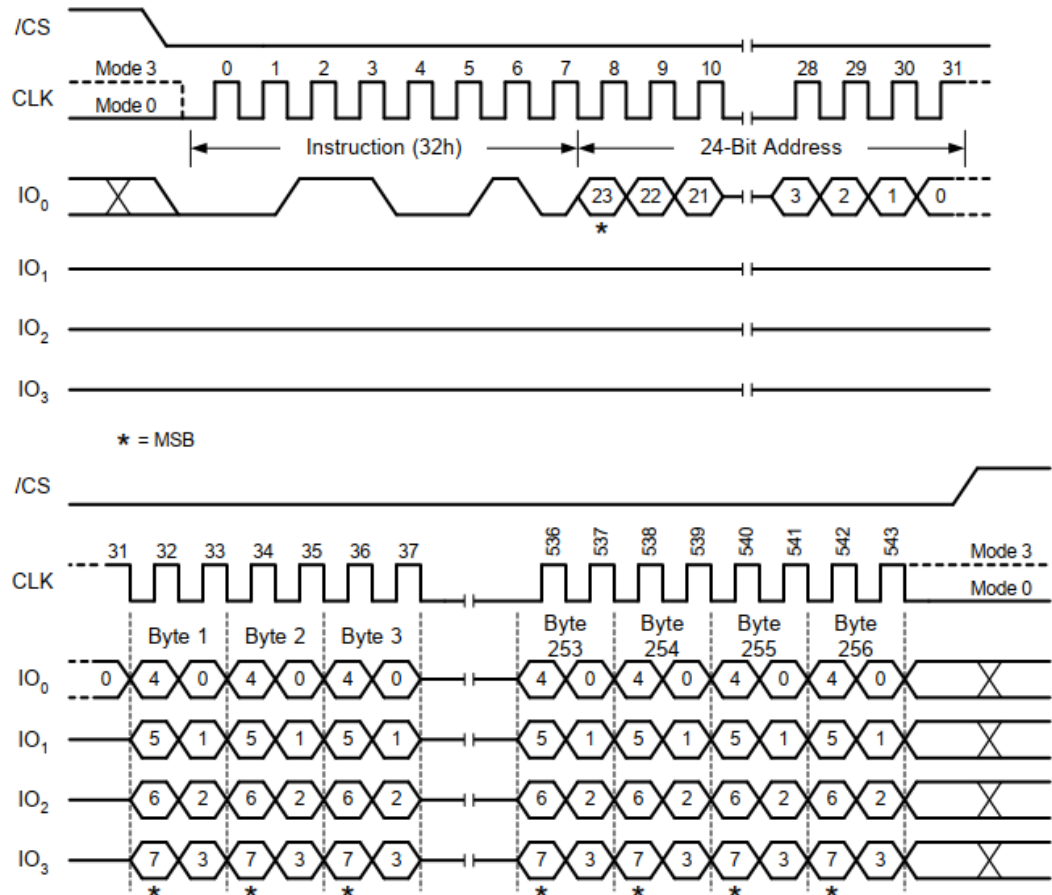
f) 设置 SPI 命令寄存器(0x24)值为 0x00000032

CMD = 0x32

注!

- 擦除命令是将 Flash 的相应区域数据位置 1，有 4 个擦除命令，4KB 扇区擦除，32KB 块擦除，64KB 块擦除，片擦除。如果被擦除区域有其他有效数据，操作者写入时需先读取被擦除区域的其他数据，及时回写。
- 写入数据量不能超过 TX FIFO 深度。

图 3-19 Quad Page Program 时序图

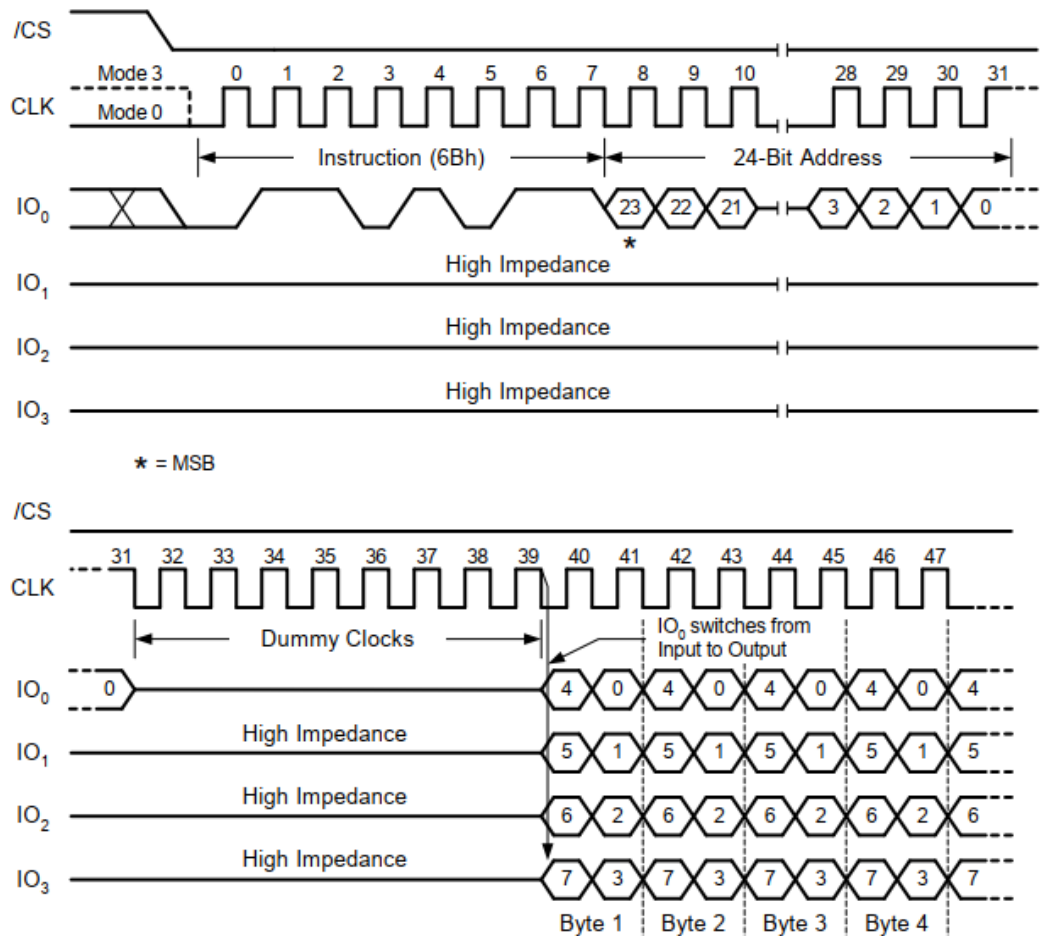


3.6.18 Fast Read Quad Output (6BH)操作过程

举例连续读取 4 个字节具体工作过程：

1. 设置 SPI 传输控制寄存器(0x20)值为 0x69800603
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) TransMode = 0x9 (Dummy, 读)
 - d) DualQuad = 0x2(Quad I/O 模式)
 - e) RdTranCnt = 3 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x0000006B
CMD = 0x6B
5. 读取 SPI 数据寄存器(0x2C)

图 3-20 Fast Read Quad Output 时序图

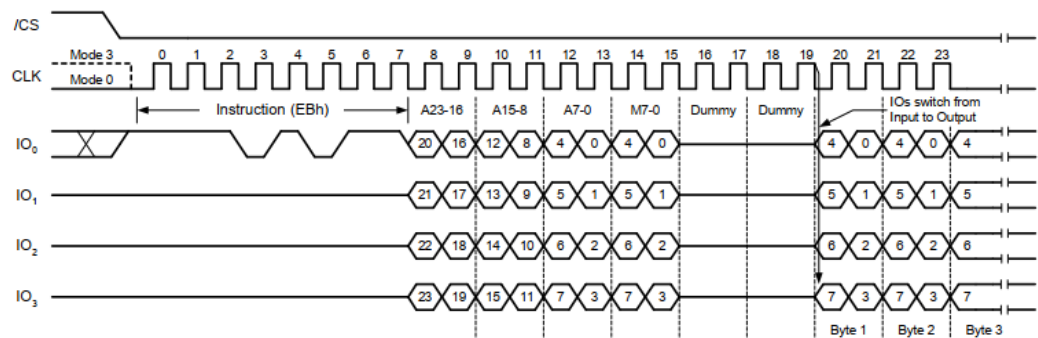


3.6.19 Fast Read Quad I/O (EBH)操作过程

举例连续读取 4 个字节具体工作过程：

1. 设置 SPI 传输控制寄存器(0x20)值为 0x79A00203
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) AddrFmt = 1
 - d) TransMode = 0x9 (Dummy, 读)
 - e) DualQuad = 0x2(Quad I/O 模式)
 - f) TokenEn = 1(Token 字节使能)
 - g) DummyCnt = 1(Dummy 数量-1)
 - h) RdTranCnt = 3 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
4. 设置 SPI 命令寄存器(0x24)值为 0x000000EB
CMD = 0xEB
5. 读取 SPI 数据寄存器(0x2C)

图 3-21 Fast Read Quad I/O 时序图



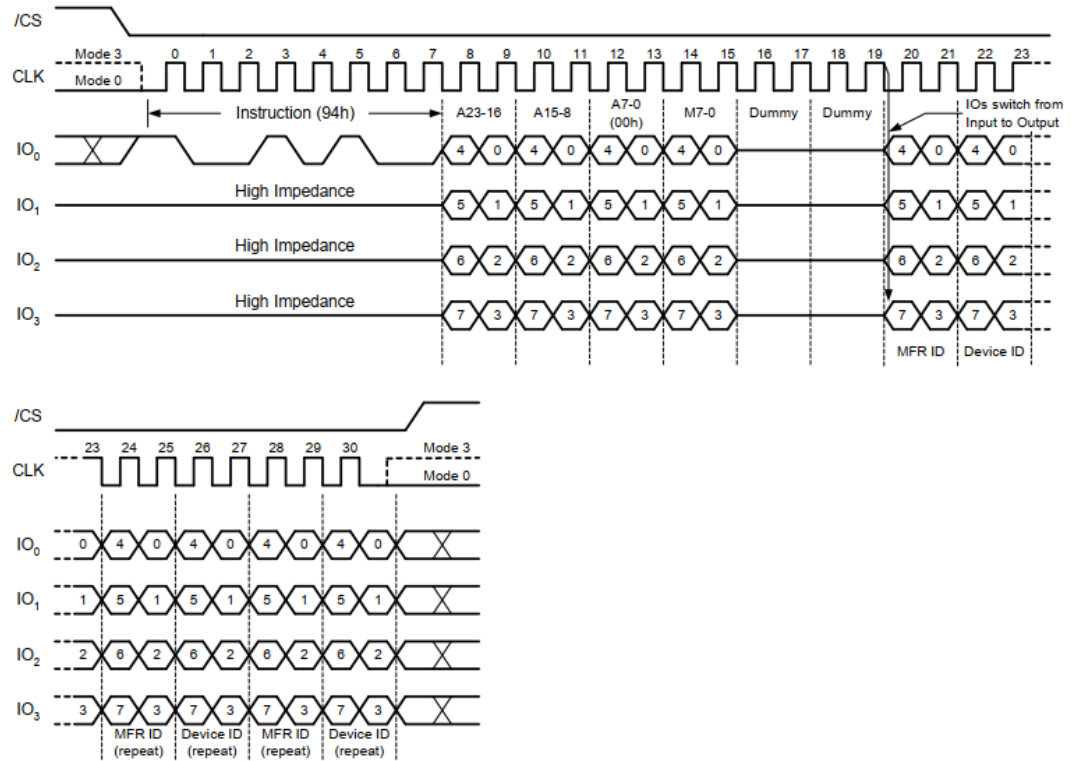
3.6.20 Read Manufacturer/Device ID Quad I/O (94H)操作过程

具体工作过程：

1. 设置 SPI 传输控制寄存器(0x20)值为 0x79A00201
 - a) CmdEn = 1
 - b) AddrEn = 1
 - c) AddrFmt = 1
 - d) TransMode = 0x9 (Dummy, 读)
 - e) DualQuad = 0x2(Quad I/O 模式)

- f) TokenEn = 1(Token 字节使能)
 - g) DummyCnt = 1(Dummy 数量-1)
 - h) RdTranCnt = 1 (传输总数-1)
2. 设置 SPI 控制寄存器(0x30)值为 0x00000002
RXFIFORST = 1
 3. 设置 SPI 地址寄存器(0x28)值为 0x00000000
ADDR = 0x00000000
 4. 设置 SPI 命令寄存器(0x24)值为 0x00000094
CMD = 0x94
 5. 读取 SPI 数据寄存器(0x2C)

图 3-22 Read Manufacturer/Device ID Quad I/O 时序图



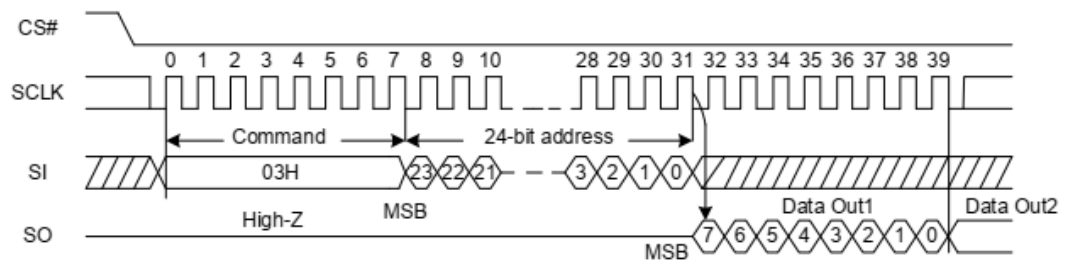
3.7 时序说明

本节介绍 Gowin SPI Flash Interface IP 的时序情况。

3.7.1 SPI 接口时序

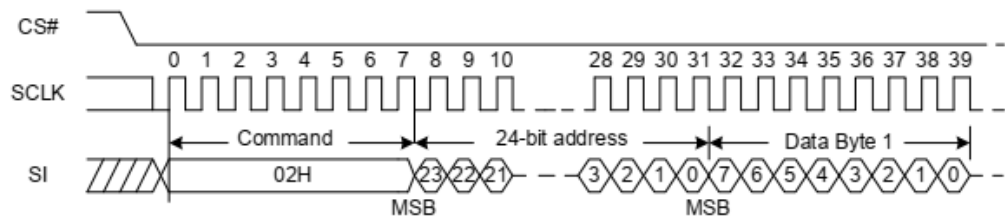
SPI Flash 读接口时序示意图如图 3-23 所示。

图 3-23 SPI Flash 读接口时序示意图



SPI Flash 写接口时序图如图 3-24 所示。

图 3-24 SPI Flash 写接口时序图



3.7.2 AHB 接口时序

图 3-25 AHB 总线无等待状态传输接口时序

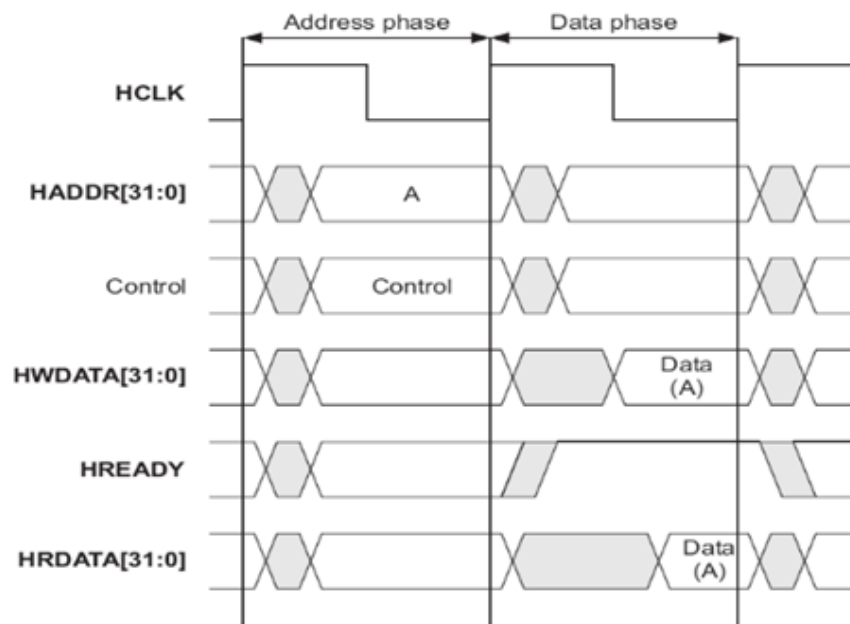
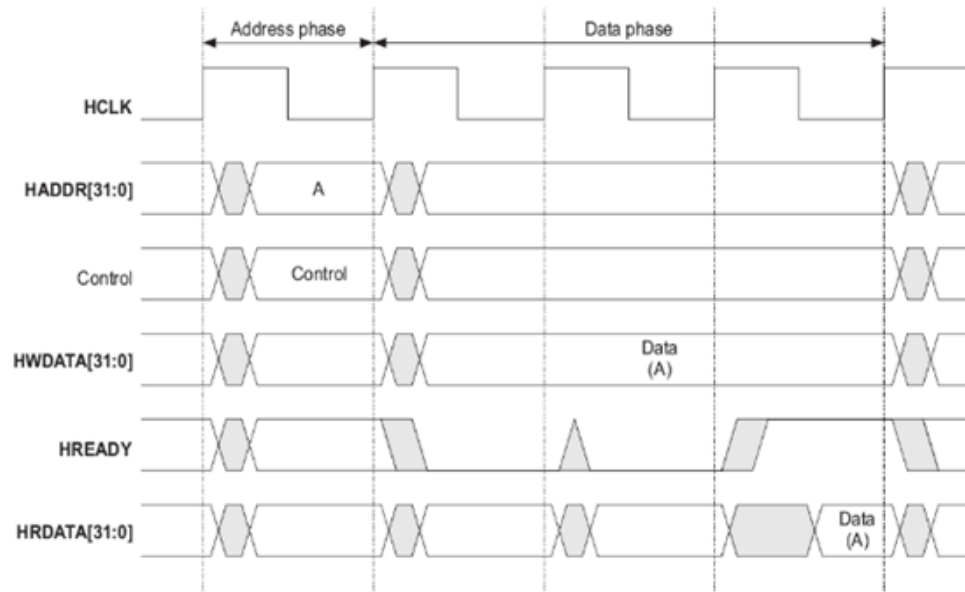


图 3-26 AHB 总线有等待状态传输接口时序



3.7.3 APB 接口时序

图 3-27 APB 总线写接口时序

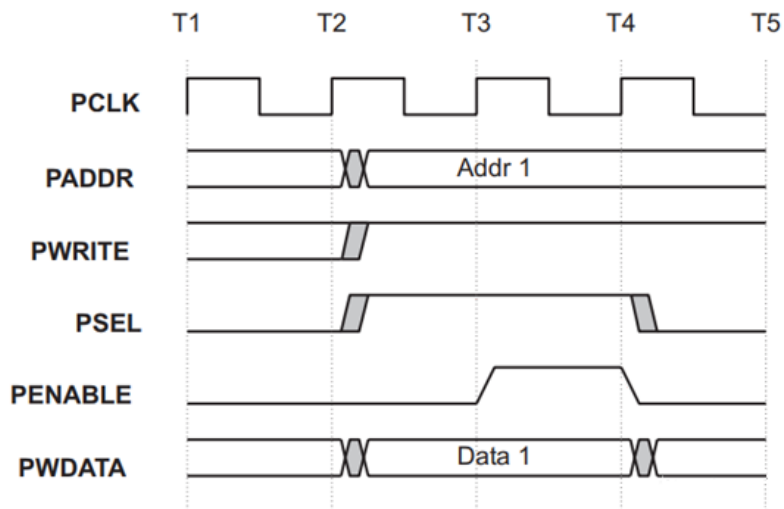
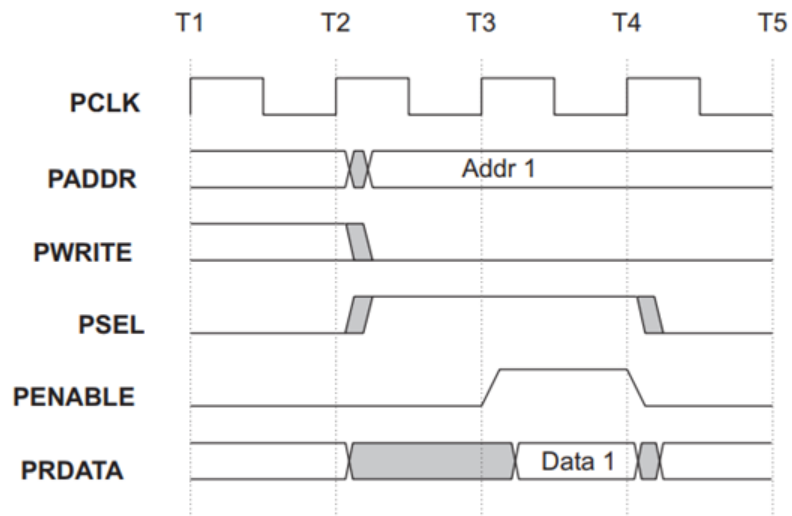


图 3-28 APB 总线读接口时序



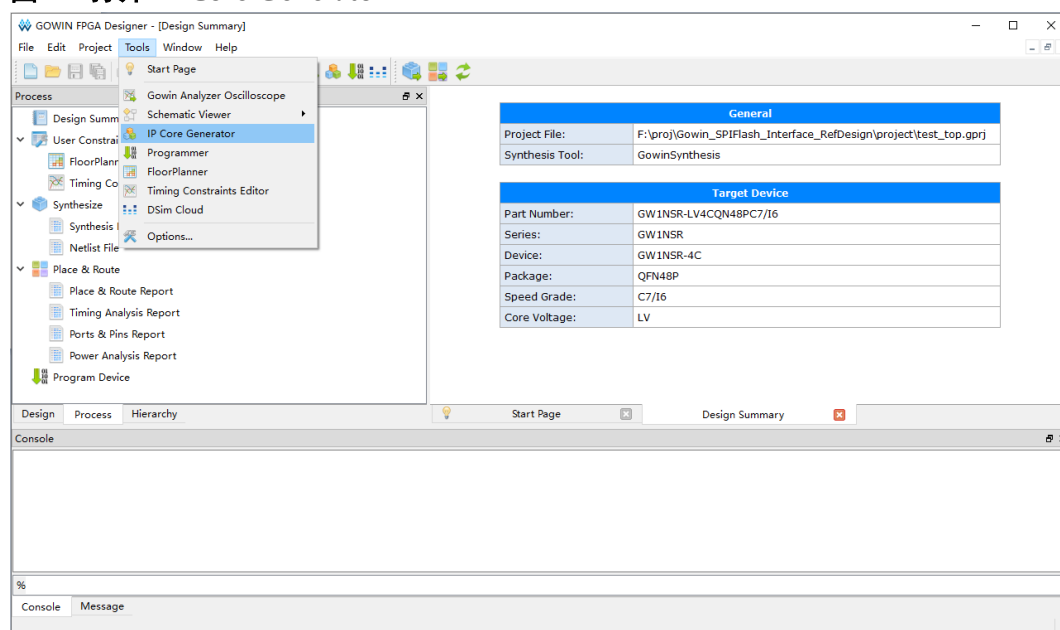
4 界面配置

用户可用高云半导体云源®软件中的 IP 内核生成器工具调用和配置高云 Gowin SPI Flash Interface IP。

1. 打开 IP Core Generator

用户建立工程后，单击左上角 Tools 选项卡，下拉单击 IP Core Generator 选项，即可打开 Gowin IP Core Generator，如图 4-1 所示。

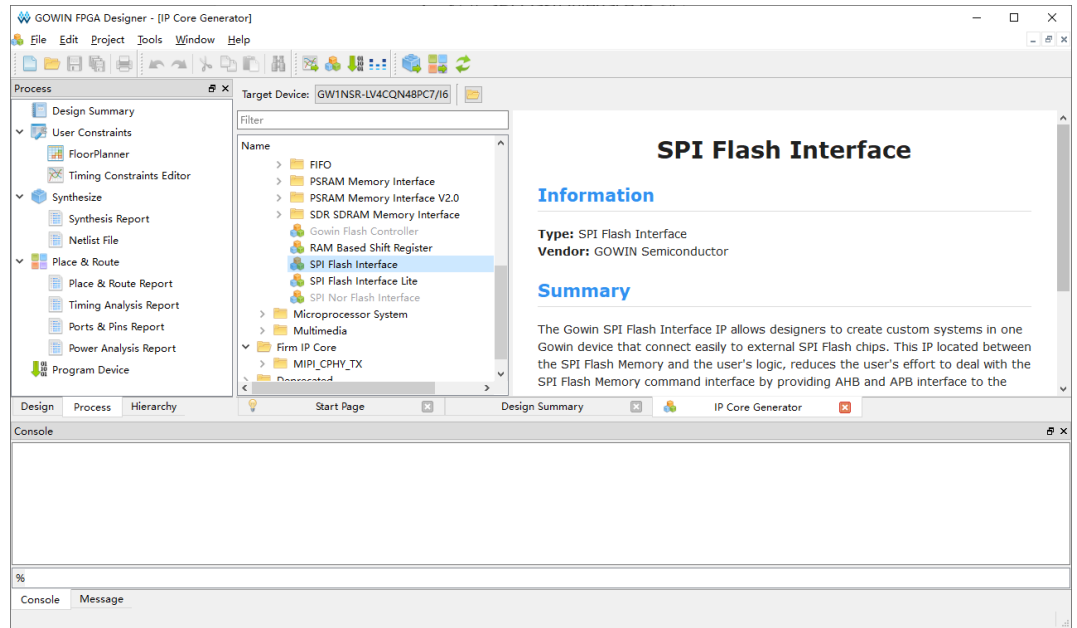
图 4-1 打开 IP Core Generator



2. 打开 SPI Flash Interface IP 核

单击 Memory Control 选项，双击 SPI Flash Interface，打开 SPI Flash Interface IP 核的配置界面，如图 4-2 所示。

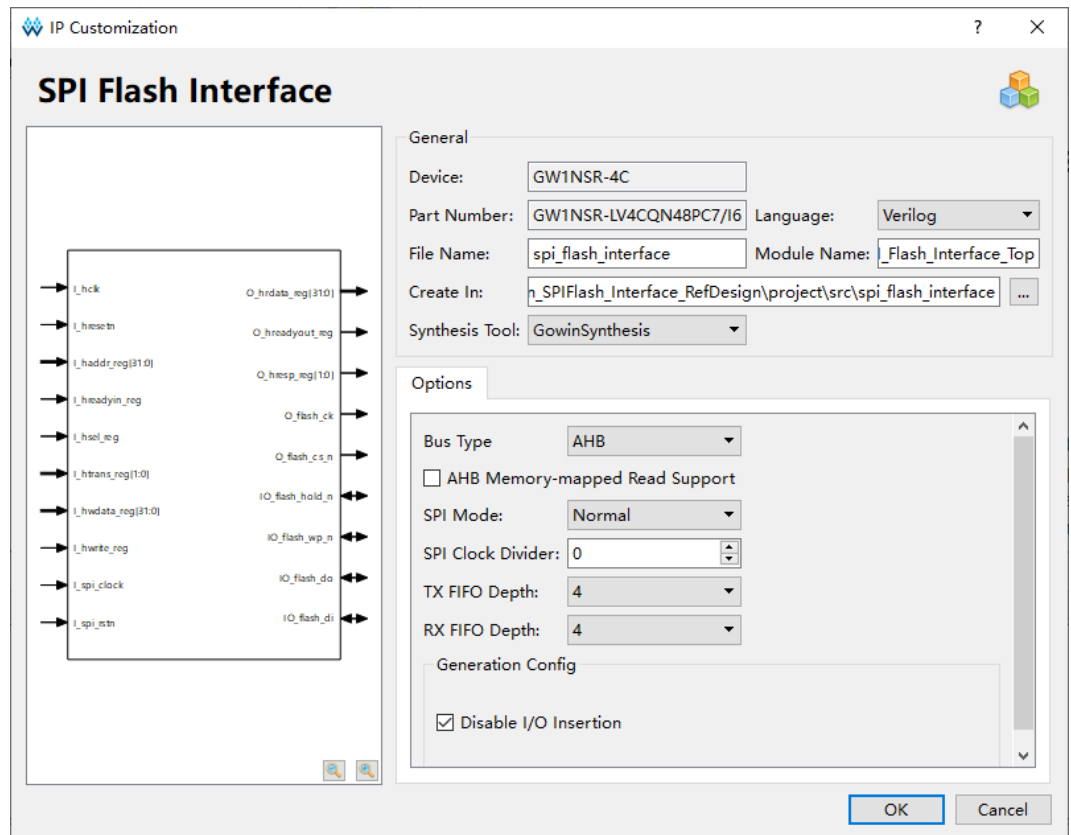
图 4-2 打开 SPI Flash Interface IP 核



3. SPI Flash Interface IP 核端口界面

配置界面左侧为 SPI Flash Interface IP 核的端口示意图，如图 4-3 所示。

图 4-3 SPI Flash Interface IP 核端口示意图



4. 配置基本信息

在配置界面的上部分是工程基本信息配置界面，本文芯片型号选择 GW1NSR-4C 为例，封装选择 QN48P。Module Name 选项后面是工程产生后顶层文件的名称，默认为“SPI_Flash_Interface_Top”，用户可自行修改。“File Name”是 IP 核文件产生的文件夹，存放 SPI Flash Interface IP 核所需文件，默认为“spi_flash_interface”，用户可自行修改路径。Create In 选项是 IP 核文件夹产生路径，默认为“\工程路径\src\spi_flash_interface”，用户可自行修改路径。

图 4-4 基本信息配置界面

The screenshot shows the 'General' configuration window with the following settings:

- Device: GW1NSR-4C
- Part Number: GW1NSR-LV4CQN48PC7/I6
- Language: Verilog
- File Name: spi_flash_interface
- Module Name: _Flash_Interface_Top
- Create In: \SPIFlash_Interface_RefDesign\project\src\spi_flash_interface
- Synthesis Tool: GowinSynthesis

5. Options 选项卡

在 Options 选项卡中，用户需要配置 SPI Flash Interface IP 总线接口，模式等参数信息。

图 4-5 Options 选项卡

The screenshot shows the 'Options' configuration window with the following settings:

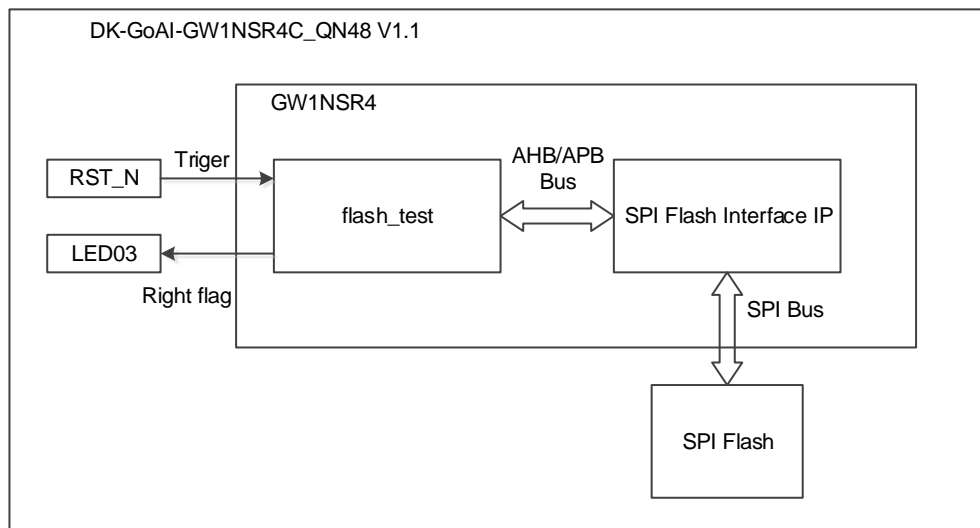
- Bus Type: AHB
- AHB Memory-mapped Read Support
- SPI Mode: Normal
- SPI Clock Divider: 0
- TX FIFO Depth: 4
- RX FIFO Depth: 4
- Generation Config:
 - Disable I/O Insertion

5 参考设计

本节主要介绍 SPI Flash Interface IP 的参考设计实例的使用方法。详细信息请参见高云半导体官网给出的 SPI Flash Interface 相关[参考设计](#)。

本参考设计以 DK-GoAI-GW1NSR4C_QN48_V1.1 开发板为例，参考设计基本框图如图 5-1 所示。DK-GoAI-GW1NSR4C_QN48_V1.1 开发板的详细信息可点击[此处](#)获取。

图 5-1 参考设计基本结构图



在参考设计中，flash_test 模块模拟 AHB 或 APB 总线主设备，当按动按键 RST_N 时，触发 flash_test 发出写和读命令，通过 SPI Flash Interface IP 对 SPI Flash 进行连续写和连续读操作，然后比较读取的数据与写入数据是否正确，如果正确，flash_test 发出 Right flag 信号给 LED03 灯进行指示，灯亮表示读写正常。

6 文件交付

Gowin SPI Flash Interface IP 交付文件主要包含三个部分，分别为：文档、设计源代码和参考设计。

6.1 文档

文件夹主要包含用户指南 PDF 文档。

表 6-1 文档列表

名称	描述
IPUG1015, Gowin SPI Flash Interface IP 用户指南	高云 SPI Flash Interface IP 用户手册，即本手册。

6.2 设计源代码（加密）

加密代码文件夹包含 Gowin SPI Flash Interface IP 的 RTL 加密代码，供 GUI 使用，以配合高云云源软件产生用户所需的 IP 核。

表 6-2 SPI Flash Interface 设计源代码列表

名称	描述
spi_flash_interface.v	IP 核顶层文件，给用户接口信息，加密。

6.3 参考设计

Gowin SPI Flash Interface RefDesign 文件夹主要包含 Gowin SPI Flash Interface IP 的网表文件，用户参考设计，约束文件、顶层文件及工程文件夹等。

表 6-3 RefDesign 文件夹内容列表

名称	描述
flash_test_top.v	参考设计的顶层模块
flash_test.v	测试激励产生模块
test_top.cst	工程物理约束文件
test_top.sdc	工程时序约束文件
spi_flash_interface	SPI Flash Interface IP 文件夹

