



Gowin 闪存资源(User Flash) 用户指南

UG295-1.0, 2020-08-24

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 |
|------------|-----|-------|
| 2020/08/24 | 1.0 | 初始版本。 |

目录

| | |
|----------------------|-----------|
| 目录 | i |
| 图目录 | ii |
| 表目录 | iii |
| 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 | 1 |
| 2 概述 | 2 |
| 3 原语介绍 | 3 |
| 3.1 FLASH96K | 3 |
| 3.2 FLASH64KZ | 10 |
| 3.3 FLASH64K | 15 |
| 3.4 FLASH128K | 21 |
| 3.5 FLASH256K | 27 |
| 3.6 FLASH608K | 32 |
| 4 IP 调用 | 38 |

图目录

| | |
|--|----|
| 图 3-1 FLASH96K 端口示意图 | 4 |
| 图 3-2 读操作模式 | 9 |
| 图 3-3 写入页锁存模式 | 9 |
| 图 3-4 清除页锁存模式 | 9 |
| 图 3-5 高电平周期 | 10 |
| 图 3-6 FLASH64KZ 端口示意图 | 11 |
| 图 3-7 读操作时序 | 14 |
| 图 3-8 写操作时序 | 15 |
| 图 3-9 擦除操作时序 | 15 |
| 图 3-10 FLASH64K 端口示意图 | 16 |
| 图 3-11 读操作时序 | 20 |
| 图 3-12 写操作时序 | 20 |
| 图 3-13 擦除操作时序 | 20 |
| 图 3-14 FLASH128K 端口示意图 | 21 |
| 图 3-15 读操作模式 | 26 |
| 图 3-16 写操作模式 | 26 |
| 图 3-17 页擦除模式 | 27 |
| 图 3-18 模块擦除模式 | 27 |
| 图 3-19 FLASH256K 端口示意图 | 28 |
| 图 3-20 读操作时序 | 31 |
| 图 3-21 写操作时序 | 32 |
| 图 3-22 擦除操作时序 | 32 |
| 图 3-23 FLASH608K 端口示意图 | 33 |
| 图 3-24 读操作时序 | 37 |
| 图 3-25 写操作时序 | 37 |
| 图 3-26 擦除操作时序 | 37 |
| 图 4-1 User Flash 的 IP Customization 窗口结构 | 39 |

表目录

| | |
|-------------------------|----|
| 表 1-1 术语、缩略语 | 1 |
| 表 3-1 适用器件 | 3 |
| 表 3-2 端口介绍 | 4 |
| 表 3-3 输出位宽选择 | 5 |
| 表 3-4 输入位宽选择 | 5 |
| 表 3-5 操作模式选择 | 5 |
| 表 3-6 时序参数 | 8 |
| 表 3-7 端口介绍 | 11 |
| 表 3-8 用户模式真值表 | 11 |
| 表 3-9 时序参数 | 13 |
| 表 3-10 端口介绍 | 16 |
| 表 3-11 用户模式真值表 | 17 |
| 表 3-12 时序参数 | 18 |
| 表 3-13 端口介绍 | 22 |
| 表 3-14 用户数据闪存地址映射 | 22 |
| 表 3-15 用户信息闪存地址映射 | 22 |
| 表 3-16 操作模式命令表 | 23 |
| 表 3-17 时序参数 | 25 |
| 表 3-18 端口介绍 | 28 |
| 表 3-19 用户模式真值表 | 29 |
| 表 3-20 时序参数 | 30 |
| 表 3-21 端口介绍 | 33 |
| 表 3-22 用户模式真值表 | 34 |
| 表 3-23 时序参数 | 35 |

1 关于本手册

1.1 手册内容

本档介绍了用户闪存资源的功能、原语定义及使用方法。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可下载、查看以下相关文档：[IPUG901](#)，Gowin Flash Controller IP 用户指南。

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

表 1-1 术语、缩略语

| 术语、缩略语 | 全称 | 含义 |
|---------|-------------------------------|----------|
| FPGA | Field Programmable Gate Array | 现场可编程门阵列 |
| IP Core | Intellectual Property Core | 知识产权核 |

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

Gowin 小蜜蜂[®]家族 FPGA 产品提供用户闪存资源 (User Flash)。不同系列器件支持不同容量大小的 Flash，包括 FLASH96K、FLASH64K、FLASH64KZ、FLASH128K、FLASH256K 和 FLASH608K。

3 原语介绍

用户闪存资源相关的原语与适用器件的对应关系如表 3-1 所示。

表 3-1 适用器件

| 原语 | 适用器件 |
|-----------|---|
| FLASH96K | GW1N-1, GW1N-1S, GW1NR-1。 |
| FLASH64KZ | GW1NZ-1 |
| FLASH64K | GW1NZ-ZV1FN32I2, GW1NZ-ZV1FN32I3, GW1NZ-ZV1CS16I2, GW1NZ-ZV1CS16I3, GW1NZ-ZV1FN32FI2, GW1NZ-ZV1FN32FI3, GW1NZ-ZV1FN32FES。 |
| FLASH128K | GW1NS-2, GW1NS-2C, GW1NSE-2C, GW1NSR-2, GW1NSR-2C。 |
| FLASH256K | GW1N-4, GW1N-4B, GW1NR-4, GW1NR-4B, GW1NRF-4B, GW1NS-4, GW1NS-4C, GW1NSER-4C, GW1NSR-4, GW1NSR-4C, GW1N-4C, GW1NR-4C。 |
| FLASH608K | GW1N-9, GW1N-9C, GW1NR-9, GW1NR-9C, GW1NR-9, GW1NR-9C。 |

注！

FLASH64K 的适用器件不支持 FLASH64KZ。

3.1 FLASH96K

原语介绍

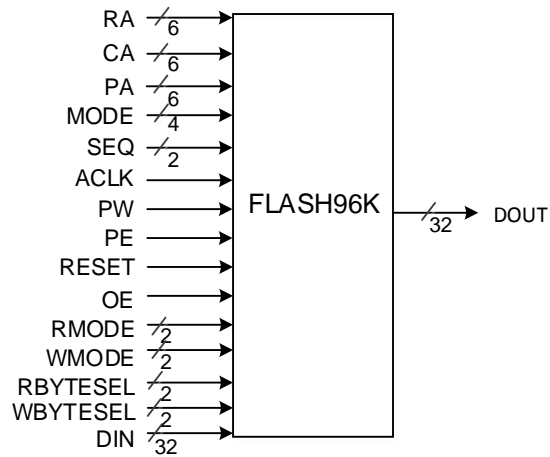
FLASH96K（96Kbit User Flash）的存储空间为 96Kbit。寄存器的宽度和深度是固定的，不可对其进行配置，其宽度为 4Byte（32 bit），地址深度为 3K，具有非易失性和断电保存功能，但不支持初始值功能。

FLASH96K 具有以下特性：

- 100,000 次写寿命周期
- 超过 10 年的数据保存能力(+85℃)
- 可选的数据输入输出位宽 8/16/32
- 页存储空间：256 Bytes
- 3μA 旁路电流
- 页写入时间：8.2ms

端口示意图

图 3-1 FLASH96K 端口示意图



端口介绍

表 3-2 端口介绍

| 端口 | I/O | 描述 |
|---------------|--------|---------------------|
| DOUT[31:0] | Output | 数据输出总线 |
| DIN[31:0] | Input | 数据输入总线 |
| RA[5:0] | Input | 行地址总线，用于选择存储单元的某一行。 |
| CA[5:0] | Input | 列地址总线，用于选择存储单元的某一列。 |
| PA[5:0] | Input | 用于选择页锁存地址的某一列 |
| MODE[3:0] | Input | 选择操作模式 |
| SEQ[1:0] | Input | 控制操作顺序 |
| ACLK | Input | 读写操作同步时钟 |
| PW | Input | 页锁存数据时钟输入 |
| RESET | Input | 复位信号，高电平有效 |
| PE | Input | 电荷泵使能 |
| OE | Input | 数据输出使能 |
| RMODE[1:0] | Input | 读数据位宽控制 |
| WMODE[1:0] | Input | 写数据位宽控制 |
| RBYTESEL[1:0] | Input | 读数据字节选择 |
| WBYTESEL[1:0] | Input | 写数据字节选择 |

配置模式

用户可以通过读写模式及读写字节选择信号改变数据的输入输出位宽，数据位宽与控制信号的对应关系如表 3-3 和表 3-4 所示。

表 3-3 输出位宽选择

| RMOD[1:0] | RBYTESEL | | DOUT | | | |
|-----------|----------|-----|---------|---------|--------|-------|
| | [1] | [0] | [31:24] | [23:16] | [15:8] | [7:0] |
| 00 | √ | √ | × | × | × | √ |
| 01 | √ | × | × | × | √ | √ |
| 1X | × | × | √ | √ | √ | √ |

表 3-4 输入位宽选择

| WMOD[1:0] | WBYTESEL | | DIN | | | |
|-----------|----------|-----|---------|---------|--------|-------|
| | [1] | [0] | [31:24] | [23:16] | [15:8] | [7:0] |
| 00 | √ | √ | × | × | × | √ |
| 01 | √ | × | × | × | √ | √ |
| 1X | × | × | √ | √ | √ | √ |

注!

“√”表示有效输入，“×”表示无效输入。

操作模式

用户可以设置 MODE[3:0]的值选择不同的操作模式,具体如表 3-5 所示。

表 3-5 操作模式选择

| MODE[3:0] | 描述 |
|-----------|-------------------|
| 0000 | 普通读操作和页锁存数据写入操作 |
| 0001 | 将预编程置位,写操作开始后自动清零 |
| 0100 | 清除页锁存数据 |
| 1000 | 页(或行)擦除 |
| 1100 | 页(或行)写操作 |

● 读操作

MODE 设置为“0000”后, ACLK 上升沿来到时进入读操作模式。读操作模式下需要保持 SEQ[1:0]的值为“00”。满足数据获取时间($\leq 38\text{ns}$)后,数据将会出现在输出管脚 DOUT。

● 写操作

用户闪存模块的写操作包含 5 步:

1. 清除页锁存数据;
2. 将数据写入页锁存;
3. 将选中的存储单元预编程虚拟的“1”;
4. 擦除选中的存储单元;
5. 将页锁存中的数据写到存储单元。

擦除存储单元后,数据变为“0”;写操作存储单元后,数据变为“1”。存储单元的“0”可以通过写操作变为“1”,但是“1”不能通过写操作变为“0”,因此,新的写入操作之前应先进行擦除。

- 写入页锁存

可以将页锁存看做是一块缓存将要写入 Flash 数据的 SRAM。写入页锁存的操作由 PW 信号控制，与 ACLK 无关。PA (Page Address) 信号指定要写入页锁存的地址。

写入页锁存操作前应先进行数据擦除。页锁存数据逐一写入，将 MODE 值设置为“0000”，SEQ[1:0]设置为“00”。页锁存的写入和数据的读取操作是完全独立的。

- 清除页锁存

与写入页锁存操作不同，清除页锁存操作是由 ACLK 控制的。MODE 设置为“0100”后，ACLK 上升沿来到时进入清除页锁存模式，此模式下 SEQ[1:0]需要保持为“00”，页锁存数据在一个 ACLK 周期内被清除。

- 擦除和写

擦除和写操作需要将 SEQ 的值按照 1> 2> 3> 0 的顺序走一遍，这些操作需要毫秒级的时间。一次擦除操作后禁止向同一页写两次。

擦除和写操作前需要通过预编程操作将选中的存储单元写入虚拟的“1”，预编程操作首先需要将 PEP (pre-program) 置位 (MODE“0001”)，然后在高电平期间写 (MODE“1100”) 选中的区域，这个过程需持续上百微秒。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体可参考第 4 章 IP 调用。

Verilog 例化:

```
FLASH96K flash96k_inst(
    .RA(ra[5:0]),
    .CA(ca[5:0]),
    .PA(pa[5:0]),
    .MODE(mode[3:0]),
    .SEQ(seq[1:0]),
    .ACLK(aclk),
    .PW(pw),
    .RESET(reset),
    .PE(pe),
    .OE(oe),
    .RMODE(rmode[1:0]),
    .WMODE(wmode[1:0]),
    .RBYTESEL(rbytesel[1:0]),
    .WBYTESEL(wbytesel[1:0]),
    .DIN(din[31:0]),
```

```

        .DOUT(dout[31:0])
    );

```

Vhdl 例化:

```

COMPONENT FLASH96K
    PORT(
        RA:IN std_logic_vector(5 downto 0);
        CA:IN std_logic_vector(5 downto 0);
        PA:IN std_logic_vector(5 downto 0);
        MODE:IN std_logic_vector(3 downto 0);
        SEQ:IN std_logic_vector(1 downto 0);
        ACLK:IN std_logic;
        PW:IN std_logic;
        RESET:IN std_logic;
        PE:IN std_logic;
        OE:IN std_logic;
        RMODE:IN std_logic_vector(1 downto 0);
        WMODE:IN std_logic_vector(1 downto 0);
        RBYTESEL:IN std_logic_vector(1 downto 0);
        WBYTESEL:IN std_logic_vector(1 downto 0);
        DIN:IN std_logic_vector(31 downto 0);
        DOUT:OUT std_logic_vector(31 downto 0)
    );
END COMPONENT;
 uut: FLASH96K
    PORT MAP (
        RA=>ra,
        CA=>ca,
        PA=>pa,
        MODE=>mode,
        SEQ=>seq,
        RESET=>reset,
        ACLK=>aclk,
        PW=>pw,
        PE=>pe,
        OE=>oe,

```

```

        RMODE=>rmode,
        WMODE=>wmode,
        RBYTESEL=>rbytesel,
        WBYTESEL=> wbytesel,
        DIN=>din,
        DOUT=>dout
    );

```

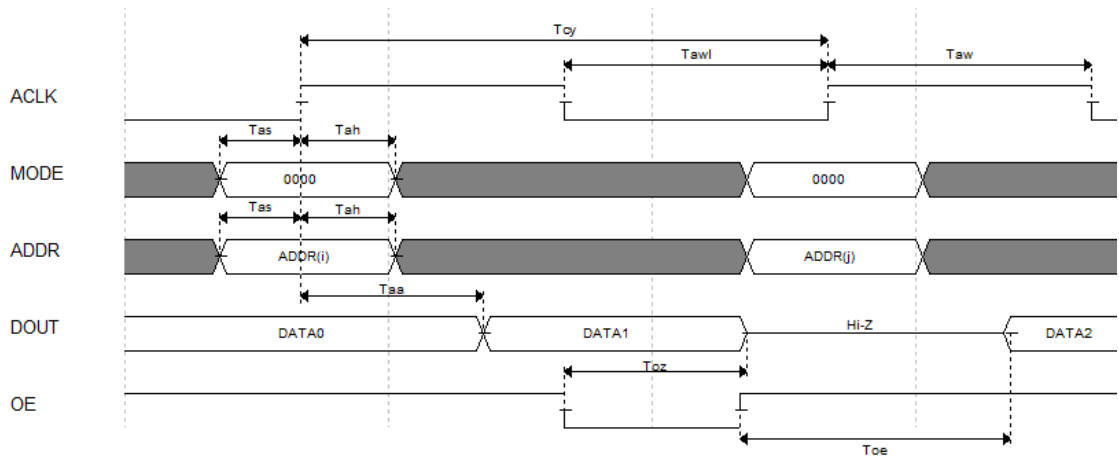
时序参数

表 3-6 时序参数

| 参数 | 描述 | 规格 | | | 单位 |
|------|-------------------|-----|-----|-----|----|
| | | 最小值 | 经典值 | 最大值 | |
| Taa | 数据获取时间 | - | - | 38 | ns |
| Tcy | 读周期 | 43 | - | - | ns |
| Taw | ACLK 高电平时间 | 10 | - | - | ns |
| Tawl | ACLK 低电平时间 | 10 | - | - | ns |
| Tas | 建立时间 | 3 | - | - | ns |
| Tah | 保持时间 | 3 | - | - | ns |
| Toz | OE 拉低到高阻态 | - | - | 2 | ns |
| Toe | OE 拉高到 DOUT | - | - | 2 | ns |
| Twcy | 写周期 | 40 | - | - | ns |
| Tpw | PW 高电平时间 | 16 | - | - | ns |
| Tpwl | PW 低电平时间 | 16 | - | - | ns |
| Tpas | 页地址建立时间 | 3 | - | - | ns |
| Tpah | 页地址保持时间 | 3 | - | - | ns |
| Tds | 数据建立时间 | 16 | - | - | ns |
| Tdh | 数据保持时间 | 3 | - | - | ns |
| Ts0 | SEQ0 周期 | 6 | - | - | μs |
| Ts1 | SEQ1 周期 | 15 | - | - | μs |
| Ts2p | ACLK 到 PE 上升沿建立时间 | 5 | - | 10 | μs |
| Ts3 | SEQ3 周期 | 5 | - | 10 | μs |
| Tps3 | PE 下降沿到 ACLK 建立时间 | 60 | - | - | μs |
| Tpe | MODE=1000 擦除时间 | 5.7 | 6 | 6.3 | ms |
| | MODE=1100 写操作时间 | 1.9 | 2 | 2.1 | ms |
| | MODE=11xx 预编程时间 | 190 | 200 | 210 | us |

时序图

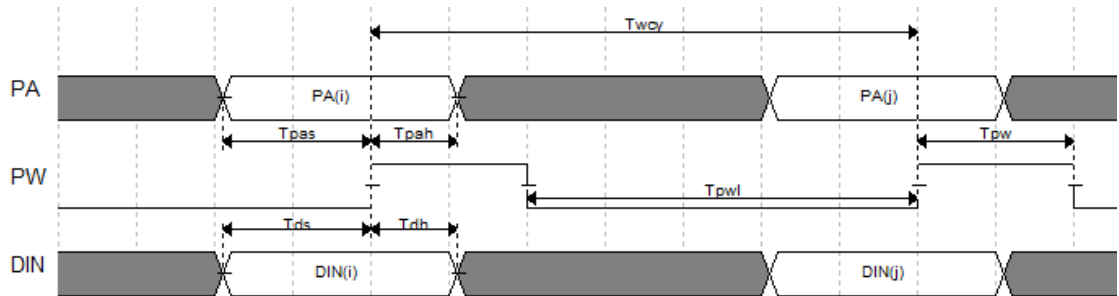
图 3-2 读操作模式



注！

读操作周期 SEQ=0, ADDR 信号包含 RA, CA, RMOD, RBYTESEL。

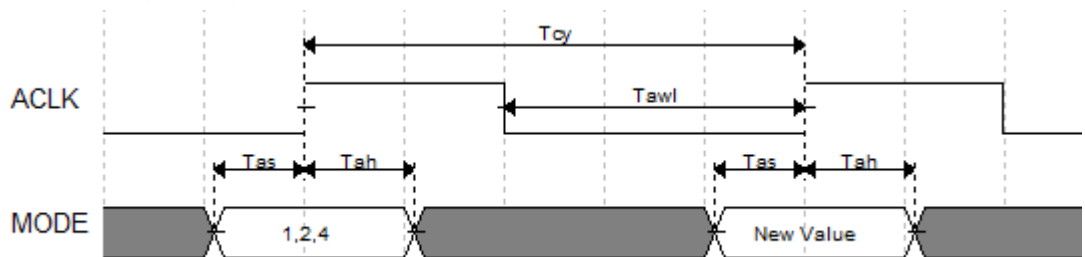
图 3-3 写入页锁存模式



注！

写入页锁存周期 SEQ=0, MODE=0000。

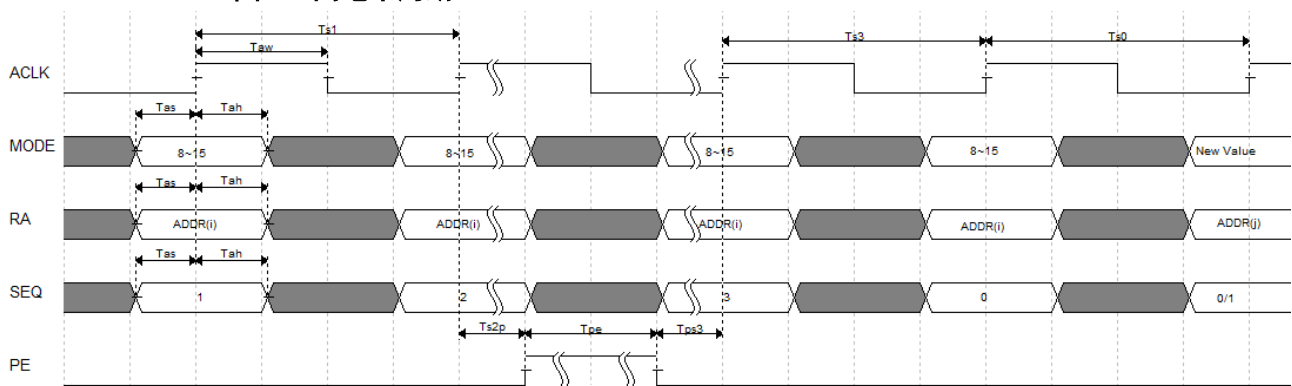
图 3-4 清除页锁存模式



注！

预编程 PEP 置位和将数据写入所有页的时序与清除页锁存模式时序参数相同，只是 MODE 值不同。

图 3-5 高电平周期



3.2 FLASH64KZ

原语介绍

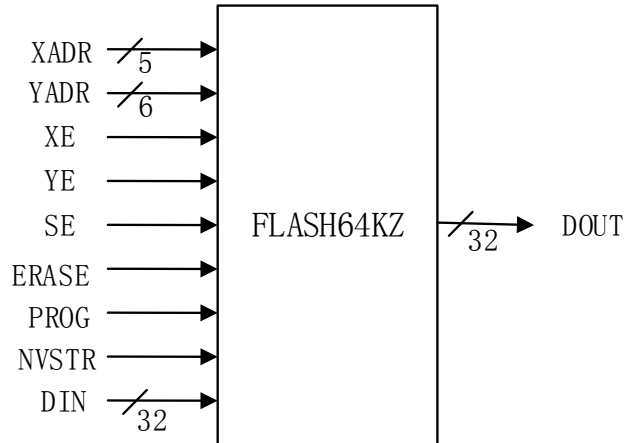
FLASH64KZ（64Kbit User Flash）的存储空间为 64Kbit。寄存器的宽度和深度是固定的，不可对其进行配置，具有非易失性和断电保存功能，但不支持初始值功能。

FLASH64KZ 具有以下特征：

- 10,000 次写寿命周期
- 容量：64K bits
- 超过 10 年的数据保存能力(+85°C)
- 支持页擦除：2,048 字节
- 快速页擦除/写操作
- 时钟频率：40MHz
- 写操作时间：≤16μs
- 页擦除时间：≤120ms
- 电流
 - 读操作：2.19mA/25ns (V_{CC}) & 0.5mA/25ns (V_{CCX})(MAX)
 - 写操作/擦除操作：12/12mA (MAX)

端口示意图

图 3-6 FLASH64KZ 端口示意图



端口介绍

表 3-7 端口介绍

| 端口 | I/O | 描述 |
|------------|--------|--------------------------------|
| DOUT[31:0] | Output | 数据输出总线 |
| DIN[31:0] | Input | 数据输入总线 |
| XADR[4:0] | Input | X 地址总线，用于选择一页存储单元中的某一行。 |
| YADR[5:0] | Input | Y 地址总线，用于选择一行存储单元中的某一列。 |
| XE | Input | X 地址使能信号，当 XE 为 0 时，所有行地址均不使能。 |
| YE | Input | Y 地址使能信号，当 YE 为 0 时，所有列地址均不使能。 |
| SE | Input | 检测放大器使能信号，高电平有效。 |
| ERASE | Input | 擦除信号，高电平有效。 |
| PROG | Input | 写信号，高电平有效。 |
| NVSTR | Input | Flash 数据存储信号，高电平有效。 |

配置模式

GW1NZ 系列 FPGA 产品的用户闪存分为通用模式和低功耗模式两种，FLASH64KZ 为通用模式用户闪存。

FLASH64KZ 默认状态为打开，器件上电后可进行正常操作，如擦除/读/写操作，不支持切换到关闭状态。

操作模式

表 3-8 用户模式真值表

| 模式 | XE | YE | SE | PROG | ERASE | NVSTR |
|-------|----|----|----|------|-------|-------|
| 读模式 | H | H | H | L | L | L |
| 写模式 | H | H | L | H | L | H |
| 页擦除模式 | H | L | L | L | H | H |

注!

“H”和“L”表示高电平和低电平。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体可参考第 4 章 IP 调用。

Verilog 例化:

```
FLASH64KZ flash64kz_inst(
    .XADR(xadr[4:0]),
    .YADR(yadr[5:0]),
    .XE(xe),
    .YE(ye),
    .SE(se),
    .ERASE(erase),
    .PROG(prog),
    .NVSTR(nvstr),
    .DIN(din[31:0]),
    .DOUT(dout[31:0])
);
```

Vhdl 例化:

```
COMPONENT FLASH64KZ
    PORT(
        XADR:IN std_logic_vector(4 downto 0);
        YADR:IN std_logic_vector(5 downto 0);
        XE:IN std_logic;
        YE:IN std_logic;
        SE:IN std_logic;
        ERASE:IN std_logic;
        PROG:IN std_logic;
        NVSTR:IN std_logic;
        DIN:IN std_logic_vector(31 downto 0);
        DOUT:OUT std_logic_vector(31 downto 0)
    );
END COMPONENT;
 uut: FLASH64KZ
    PORT MAP (
```

XADR=>xadr,
 YADR=>yadr,
 XE=>xe,
 YE=>ye,
 SE=>se,
 ERASE=>erase,
 PROG=>prog,
 NVSTR=>nvstr,
 DIN=>din,

DOUT=>dout

);

时序参数

表 3-9 时序参数

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|----------------|-----|------------|-----|-----|---------|
| 访问时间 | WC1 | T_{acc} | - | 25 | ns |
| | TC | | - | 22 | ns |
| | BC | | - | 21 | ns |
| | LT | | - | 21 | ns |
| | WC | | - | 25 | ns |
| 写/擦除到数据存储建立时间 | | T_{nvs} | 5 | - | μs |
| 数据存储保持时间 | | T_{nvh} | 5 | - | μs |
| 数据存储保持时间(整体擦除) | | T_{nvh1} | 100 | - | μs |
| 数据存储到写建立时间 | | T_{pgs} | 10 | - | μs |
| 写保持时间 | | T_{pgh} | 20 | - | ns |
| 写时间 | | T_{prog} | 8 | 16 | μs |
| 写准备时间 | | T_{wpr} | >0 | - | ns |
| 擦除保持时间 | | T_{whd} | >0 | - | ns |
| 控制信号到写/擦除建立时间 | | T_{cps} | -10 | - | ns |
| SE 到读操作建立时间 | | T_{as} | 0.1 | - | ns |
| SE 脉冲的高电平时间 | | T_{pws} | 5 | - | ns |
| 地址/数据建立时间 | | T_{ads} | 20 | - | ns |
| 地址/数据保持时间 | | T_{adh} | 20 | - | ns |
| 数据保持时间 | | T_{dh} | 0.5 | - | ns |
| 读模式地址保持时间 | WC1 | T_{ah} | 25 | - | ns |
| | TC | - | 22 | - | ns |
| | BC | - | 21 | - | ns |
| | LT | - | 21 | - | ns |
| | WC | - | 25 | - | ns |

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|---------------------|----|--------------|-----|-----|---------|
| SE 脉冲低电平时间 | | T_{nws} | 2 | - | ns |
| 恢复时间 | | T_{rcv} | 10 | - | μ s |
| 数据存储时间 | | T_{hv} | - | 6 | ms |
| 擦除时间 | | T_{erase} | 100 | 120 | ms |
| 整体擦除时间 | | T_{me} | 100 | 120 | ms |
| 掉电到待机模式的 Wake-up 时间 | | T_{wk_pd} | 7 | - | μ s |
| 待机保持时间 | | T_{sbh} | 100 | - | ns |
| V_{CC} 建立时间 | | T_{ps} | 0 | - | ns |
| V_{CCX} 保持时间 | | T_{ph} | 0 | - | ns |

注!

- 这些设定值可能会改变。
- 这些数值为仿真数据，在实际器件中会有改变。
- 在信号 XADR、YADR、XE 和 YE 信号有效后， T_{acc} 的开始时间为 SE 信号的上升沿。读取的数据 DOUT 被保存直到在下一有效读操作开始。
- T_{hv} 时间为写操作开始到数据下一次擦除操作之前的累积时间，同一个地址在下一擦除之前不能被写入两次；同一个存储单元在下一擦除之前不能被写入两次。这种限制是基于安全考虑的。
- 所有的波形都有 1ns 的上升沿时间和 1ns 的下降沿时间。
- 控制信号 X、YADR、XE 和 YE 信号需要至少保持 T_{acc} 的时间， T_{acc} 从 SE 的上升沿处开始。

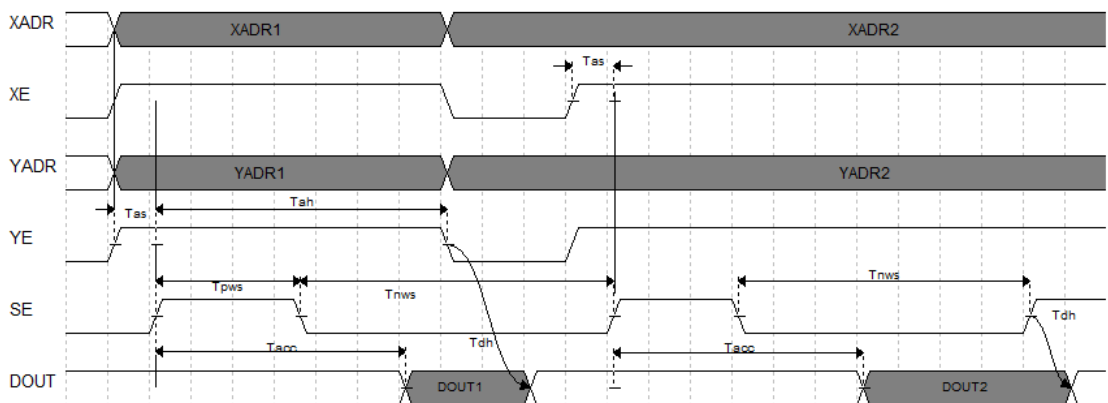
时序图**图 3-7 读操作时序**

图 3-8 写操作时序

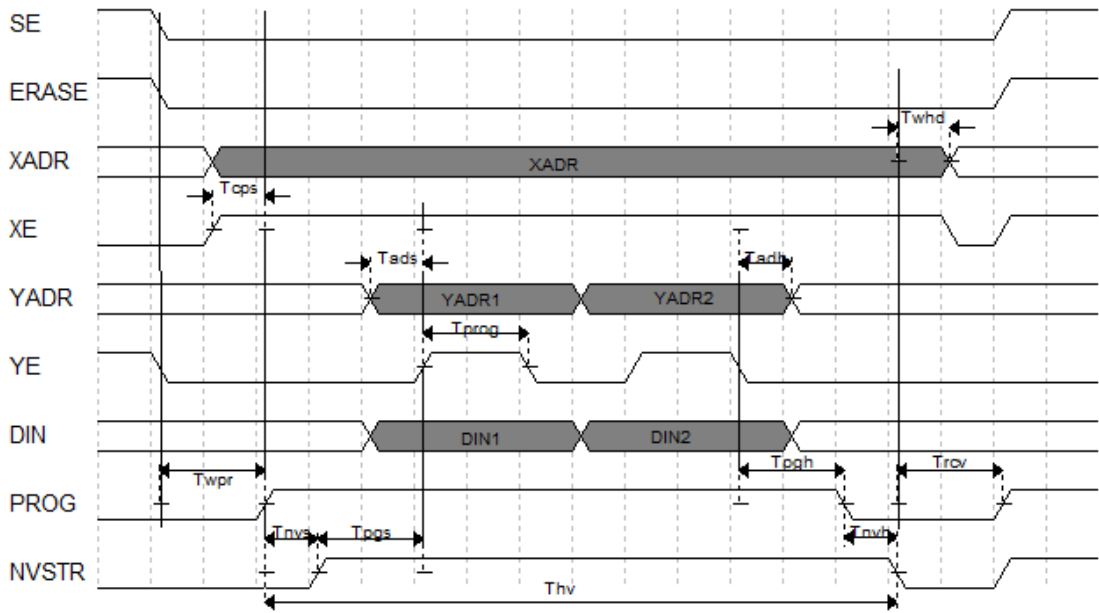
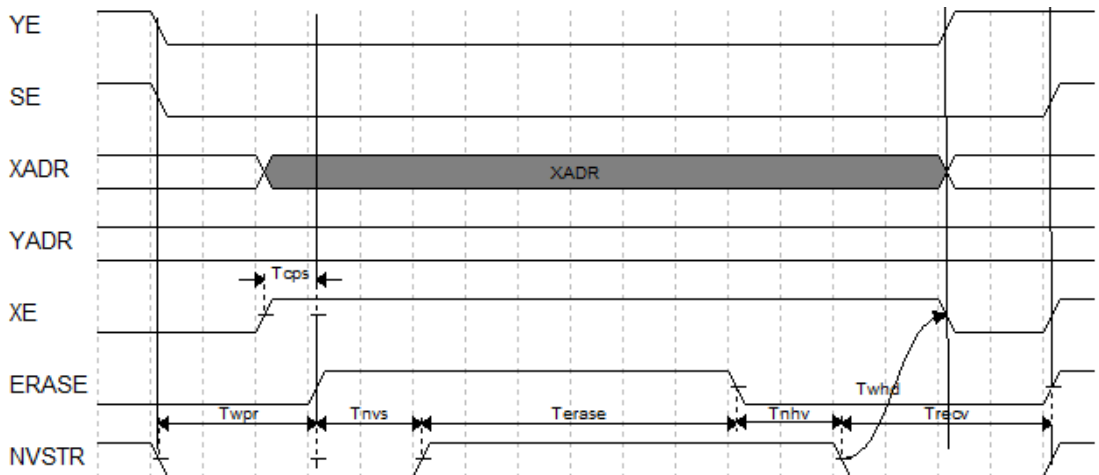


图 3-9 擦除操作时序



3.3 FLASH64K

原语介绍

FLASH64K (64Kbit User Flash) 的存储空间为 64K bit。寄存器的宽度和深度是固定的，不可对其进行配置，具有非易失性和断电保存功能，但不支持初始值功能。FLASH64K 具有睡眠模式，当信号 SLEEP 为高电平时，FLASH 进入睡眠模式。

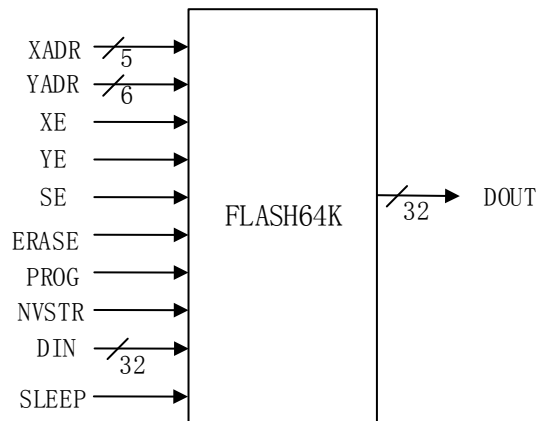
FLASH64K 具有以下特征：

- 10,000 次写寿命周期
- 容量：64K bits
- 超过 10 年的数据保存能力(+85℃)
- 支持页擦除：2,048 字节
- 快速页擦除/写操作

- 时钟频率：40MHz
- 字写操作时间：≤16μs
- 页擦除时间：≤120ms
- 电流
 - 读操作：2.19mA/25ns (V_{CC}) & 0.5mA/25ns (V_{CCX})(MAX)
 - 写操作/擦除操作：12/12mA（最大）

端口示意图

图 3-10 FLASH64K 端口示意图



端口介绍

表 3-10 端口介绍

| 端口 | I/O | 描述 |
|------------|--------|---|
| DOUT[31:0] | Output | 数据输出总线。 |
| DIN[31:0] | Input | 数据输入总线。 |
| XADR[4:0] | Input | X 地址总线，用于选择一页存储单元中的某一行。 |
| YADR[5:0] | Input | Y 地址总线，用于选择一行存储单元中的某一列。 |
| XE | Input | X 地址使能信号，当 XE 为 0 时，所有行地址均不使能。 |
| YE | Input | Y 地址使能信号，当 YE 为 0 时，所有列地址均不使能。 |
| SE | Input | 检测放大器使能信号，高电平有效。 |
| ERASE | Input | 擦除信号，高电平有效。 |
| PROG | Input | 写信号，高电平有效。 |
| NVSTR | Input | Flash 数据存储信号，高电平有效。 |
| SLEEP | Input | 低功耗用户闪存状态切换控制信号。 <ul style="list-style-type: none"> ● 高电平：打开状态 ● 低电平：关闭状态 |

配置模式

GW1NZ 系列 FPGA 产品的用户闪存分为通用模式和低功耗模式两种，

FLASH64K 为低功耗模式用户闪存。

FLASH64K 默认状态为关闭，可有效的降低功耗，用户通过控制 SLEEP 管脚可以动态切换状态，打开/关闭。切换到打开状态时，和 FLASH64KZ 一样，可进行擦除/读/写操作。

操作模式

表 3-11 用户模式真值表

| 模式 | XE | YE | SE | PROG | ERASE | NVSTR |
|-------|----|----|----|------|-------|-------|
| 读模式 | H | H | H | L | L | L |
| 写模式 | H | H | L | H | L | H |
| 页擦除模式 | H | L | L | L | H | H |

注！

“H”和“L”表示高电平和低电平。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体可参考第 4 章 IP 调用。

Verilog 例化：

```
FLASH64K flash64k_inst(
    .XADR(xadr[4:0]),
    .YADR(yadr[5:0]),
    .XE(xe),
    .YE(ye),
    .SE(se),
    .ERASE(erase),
    .PROG(prog),
    .NVSTR(nvstr),
    .DIN(din[31:0]),
    .SLEEP(sleep),
    .DOUT(dout[31:0])
);
```

Vhdl 例化：

```
COMPONENT FLASH64K
PORT(
    XADR:IN std_logic_vector(4 downto 0);
    YADR:IN std_logic_vector(5 downto 0);
    XE:IN std_logic;
    YE:IN std_logic;
```

```

        SE:IN std_logic;
        ERASE:IN std_logic;
        PROG:IN std_logic;
        NVSTR:IN std_logic;
        DIN:IN std_logic_vector(31 downto 0);
        SLEEP:IN std_logic;
        DOUT:OUT std_logic_vector(31 downto 0)
    );
END COMPONENT;
 uut: FLASH64K
    PORT MAP (
        XADR=>xadr,
        YADR=>yadr,
        XE=>xe,
        YE=>ye,
        SE=>se,
        ERASE=>erase,
        PROG=>prog,
        NVSTR=>nvstr,
        DIN=>din,
        SLEEP=>sleep,
        DOUT=>dout
    );

```

时序参数

表 3-12 时序参数

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|----------------|-----|------------|-----|-----|---------|
| 访问时间 | WC1 | T_{acc} | - | 25 | ns |
| | TC | | - | 22 | ns |
| | BC | | - | 21 | ns |
| | LT | | - | 21 | ns |
| | WC | | - | 25 | ns |
| 写/擦除到数据存储建立时间 | | T_{nvs} | 5 | - | μs |
| 数据存储保持时间 | | T_{nvh} | 5 | - | μs |
| 数据存储保持时间(整体擦除) | | T_{nvh1} | 100 | - | μs |
| 数据存储到写建立时间 | | T_{pgs} | 10 | - | μs |
| 写保持时间 | | T_{pgh} | 20 | - | ns |

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|---------------------|-----|--------------|-----|-----|---------|
| 写时间 | | T_{prog} | 8 | 16 | μs |
| 写准备时间 | | T_{wpr} | >0 | - | ns |
| 擦除保持时间 | | T_{whd} | >0 | - | ns |
| 控制信号到写/擦除建立时间 | | T_{cps} | -10 | - | ns |
| SE 到读操作建立时间 | | T_{as} | 0.1 | - | ns |
| SE 脉冲的高电平时间 | | T_{pws} | 5 | - | ns |
| 地址/数据建立时间 | | T_{ads} | 20 | - | ns |
| 地址/数据保持时间 | | T_{adh} | 20 | - | ns |
| 数据保持时间 | | T_{dh} | 0.5 | - | ns |
| 读模式地址保持时间 | WC1 | T_{ah} | 25 | - | ns |
| | TC | - | 22 | - | ns |
| | BC | - | 21 | - | ns |
| | LT | - | 21 | - | ns |
| | WC | - | 25 | - | ns |
| SE 脉冲低电平时间 | | T_{nws} | 2 | - | ns |
| 恢复时间 | | T_{rcv} | 10 | - | μs |
| 数据存储时间 | | T_{hv} | - | 6 | ms |
| 擦除时间 | | T_{erase} | 100 | 120 | ms |
| 整体擦除时间 | | T_{me} | 100 | 120 | ms |
| 掉电到待机模式的 Wake-up 时间 | | T_{wk_pd} | 7 | - | μs |
| 待机保持时间 | | T_{sbh} | 100 | - | ns |
| V_{CC} 建立时间 | | T_{ps} | 0 | - | ns |
| V_{CCX} 保持时间 | | T_{ph} | 0 | - | ns |

注!

- 这些设定值可能会改变。
- 这些数值为仿真数据，在实际器件中会有改变。
- 在信号 XADR、YADR、XE 和 YE 信号有效后， T_{acc} 的开始时间为 SE 信号的上升沿。读取的数据 DOUT 被保存直到在下一有效读操作开始。
- T_{hv} 时间为写操作开始到数据下一次擦除操作之前的累积时间，同一个地址在下一擦除之前不能被写入两次；同一个存储单元在下一擦除之前不能被写入两次。这种限制是基于安全考虑的。
- 所有的波形都有 1ns 的上升沿时间和 1ns 的下降沿时间。
- 控制信号 X、YADR、XE 和 YE 信号需要至少保持 T_{acc} 的时间， T_{acc} 从 SE 的上升沿处开始。

时序图

图 3-11 读操作时序

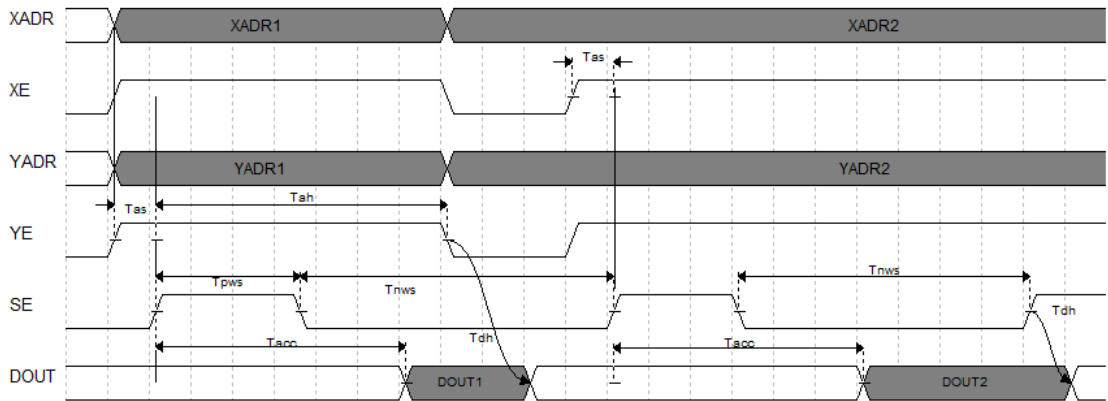


图 3-12 写操作时序

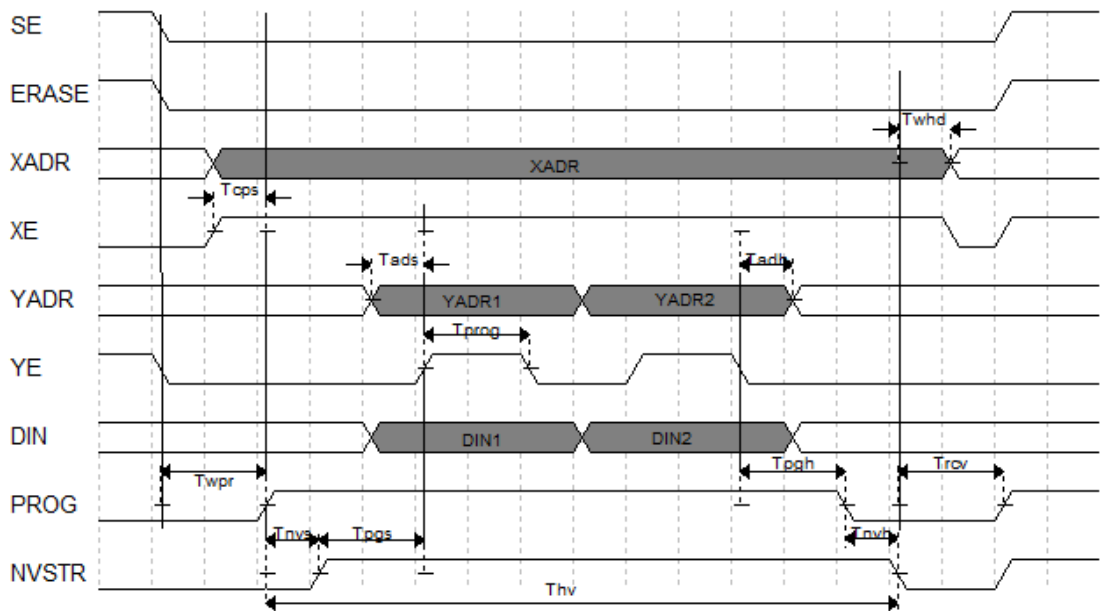
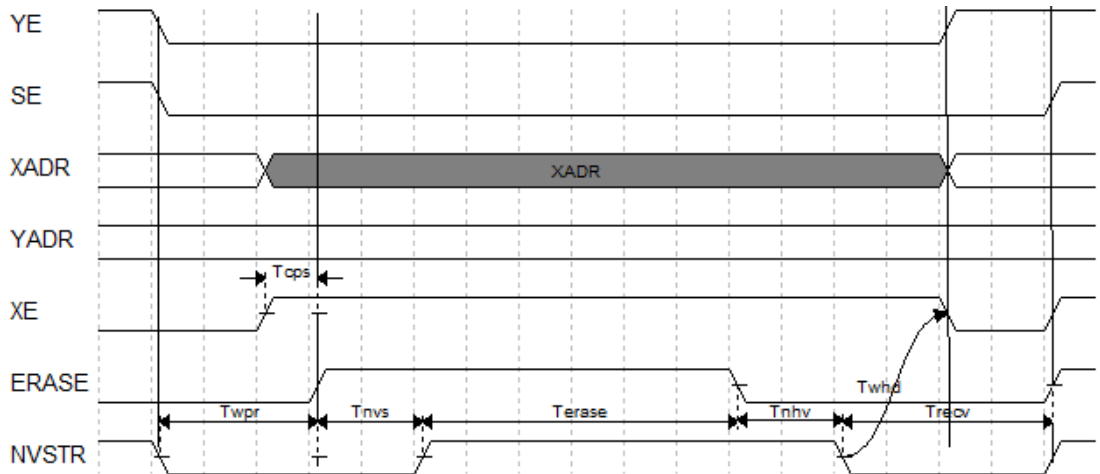


图 3-13 擦除操作时序



3.4 FLASH128K

原语介绍

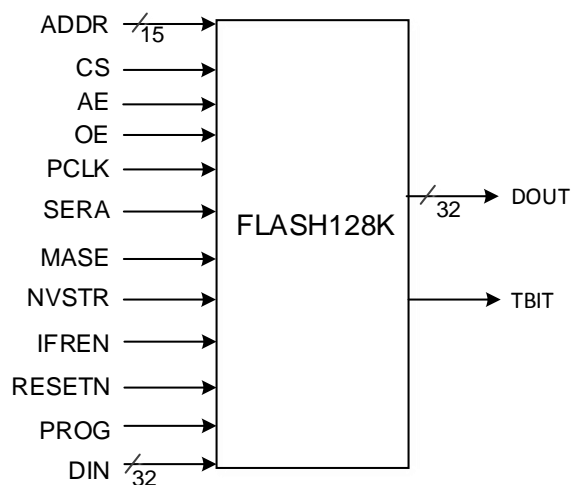
FLASH128K (128KByte Embedded Flash) 的存储空间为 128K Byte。寄存器的宽度和深度是固定的，不可对其进行配置。具有非易失性和断电保存功能，但不支持初始值功能。

FLASH128K 具有以下特征：

- 32bits 数据输入/输出
- 页架构模式
 - 一页存储空间为 128 x 32bits
 - 一共 256 页
- 快速的读，写和擦除
 - 读取时间 30ns
 - 写入时间 30us
 - 页擦除时间 2ms
 - 模块擦除时间 10ms
- 低功耗
 - IDLE 模式电流 100uA
 - 读操作电流 60uA /MHz
 - 写操作电流 2.4mA
 - 擦除操作电流 2.4mA
- 100,000 次写/擦除寿命周期
- 超过 10 年的数据保存能力

端口示意图

图 3-14 FLASH128K 端口示意图



端口介绍

表 3-13 端口介绍

| 端口 | I/O | 描述 |
|------------|--------|---------------|
| DOUT[31:0] | Output | 数据输出总线。 |
| TBIT | Output | 写/擦除完成指示信号。 |
| DIN[31:0] | Input | 数据输入总线。 |
| ADDR[14:0] | Input | 地址输入总线。 |
| CS | Input | 片选信号。 |
| AE | Input | 地址使能信号。 |
| OE | Input | 读数据使能信号。 |
| PCLK | Input | 时钟输入信号。 |
| PROG | Input | 写信号。 |
| SERA | Input | 页擦除信号。 |
| MASE | Input | 模块擦除信号。 |
| NVSTR | Input | 数据写入闪存信号。 |
| IFREN | Input | Flash 信息页面选择。 |
| RESETN | Input | 上电复位信号，低电平有效。 |

配置模式

128KB 用户闪存由 256 页组成，每页大小为 512byte，分为两行，一行包括 64 个 32bits 的数据。

表 3-14 用户数据闪存地址映射

| 页选择地址 | | | | | | | | 行选择 | 32bits 数据列地址选择 | | | | | | |
|-------|-----|-----|-----|-----|----|----|----|-----|----------------|----|----|----|----|----|--|
| A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | |
| X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 | X0 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 | |

表 3-15 用户信息闪存地址映射

| 页选择地址 | | | | | | | | 行选择 | 32bits 数据列地址选择 | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|-----|----------------|----|----|----|----|----|--|
| A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | |
| 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | X1 | X0 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 | |

操作模式

用户可以通过控制信号选择不同的操作模式，具体表 3-16 所示。

表 3-16 操作模式命令表

| MODE | CS | AE | OE | PROG | SERA | MASE | DIN | DOUT | ADDR | NVSTR |
|------|----|----|----|------|------|------|-----|------|------|-------|
| IDLE | L | L | L | L | L | L | X | Z | X | X |
| 读 | H | R | H | L | L | L | X | DOUT | ADDR | L |
| 写 | H | R | L | H | L | L | DIN | Z | ADDR | H |
| 页擦除 | H | R | L | L | H | L | X | Z | ADDR | H |
| 模块擦除 | H | R | L | L | L | H | X | Z | ADDR | H |

- 读操作

Flash 的读操作类似于 ROM 的读操作，数据读取需要满足以下条件：

AE 信号拉高，满足地址建立时间 ($\geq 5\text{ns}$)，在 AE 的上升沿，地址被锁存；

OE 读使能信号拉高，($\geq 1\text{ns}$) 数据出现在输出管脚 DOUT 上，数据读取时间为 30ns。

- 写操作

写操作前需要先对想要写入的存储空间进行擦除操作，因为在 flash 数据写入到非易失性存储空间的时候，只能是从 1 变 0，只有擦除操作才能使数据从 0 变 1。

Flash 的写操作(数据 0)类似于 SRAM 的写操作，完成写操作需要 AE、PROG 和 NVSTR 为高电平。写入的数据和地址提前准备好，满足建立时间 ($\geq 5\text{ns}$)，在 AE 的上升沿，数据和地址被锁存。AE 变为高电平后，满足 NVSTR 信号的上升沿保持时间 ($\geq 10\text{ns}$)，NVSTR 信号变为高电平，数据写入到存储空间的地址中，写入时间为 30us。

- 擦除操作

用户闪存支持页擦除和模块擦除，页擦除的空间是 512byte，模块擦除的空间是整个用户数据存储空间。当 SERA 为高电平时，有效的擦除操作是页擦除，当 MASE 为高电平时，有效的擦除操作是模块擦除。

页擦除操作：完成页擦除操作需要 AE、SERA 和 NVSTR 为高电平。擦除的地址提前准备好，满足建立时间 ($\geq 5\text{ns}$)，在 AE 的上升沿，地址被锁存。AE 变为高电平后，满足 NVSTR 信号的上升沿保持时间 ($\geq 10\text{ns}$)，NVSTR 信号变为高电平，对应地址的存储空间被擦除，数据变为 1，页擦除时间为 2ms。

模块擦除操作：完成模块擦除操作需要 AE、MASE 和 NVSTR 为高电平。擦除的地址提前准备好，满足建立时间 ($\geq 5\text{ns}$)，在 AE 的上升沿，地址被锁存。AE 变为高电平后，满足 NVSTR 信号的上升沿保持时间 ($\geq 10\text{ns}$)，NVSTR 信号变为高电平，对应地址的存储空间被擦除，数据变为 1，模块擦除时间为 10ms。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体

可参考第 4 章 IP 调用。

Verilog 例化:

```
FLASH128K flash128k_inst(  
    .ADDR(addr[14:0]),  
    .CS(cs),  
    .AE(ae),  
    .OE(oe),  
    .PCLK(pclk),  
    .PROG(prog),  
    .SERA(sera),  
    .MASE(mase),  
    .NVSTR(nvstr),  
    .IFREN(ifren),  
    .RESETN(resetn),  
    .DIN(din[31:0]),  
    .DOUT(dout[31:0]),  
    .TBIT(tbit)  
);
```

Vhdl 例化:

```
COMPONENT FLASH128K  
    PORT(  
        DIN:IN std_logic_vector(31 downto 0);  
        ADDR:IN std_logic_vector(14 downto 0);  
        CS:IN std_logic;  
        AE:IN std_logic;  
        OE:IN std_logic;  
        PCLK:IN std_logic;  
        PROG:IN std_logic;  
        SERA:IN std_logic;  
        MASE:IN std_logic;  
        NVSTR:IN std_logic;  
        IFREN:IN std_logic;  
        RESETN:IN std_logic;  
        DOUT:OUT std_logic_vector(31 downto 0);  
        TBIT:OUT std_logic;
```

```

);
END COMPONENT;
uut: FLASH128K
PORT MAP (
    DIN=>din,
    ADDR=>addr,
    CS=>cs,
    AE=>ae,
    OE=>oe,
    PCLK=>pclk,
    PROG=>prog,
    SERA=>sera,
    MASE=>mase,
    NVSTR=>nvstr,
    IFREN=>ifren,
    RESETN=>resetn,
    DOUT=>dout,
    TBIT=>tbit
);

```

时序参数

表 3-17 时序参数

| 符号 | 描述 | 规格 | | 单位 |
|-------|-----------------------|-----|-----|----|
| | | 最小值 | 最大值 | |
| tAS | 地址建立时间 | 2 | - | ns |
| tHS | 地址保持时间 | 2 | - | ns |
| tS | 写和擦除建立时间 | 5 | - | ns |
| tH | 写和擦除保持时间 | 5 | - | ns |
| tDS | 数据建立时间 | 5 | - | ns |
| tDH | 数据保持时间 | 5 | - | ns |
| tAC | 数据读取时间 | - | 30 | ns |
| tACR | | - | 80 | ns |
| tHZ | 高阻到 OE 变为低电平的时间 | 3 | - | ns |
| tAE | AE 高电平时间 | 10 | - | ns |
| tAEL | AE 低电平时间 | 10 | - | ns |
| tAAD | 读操作下 AE 到 AE 延迟时间 | 30 | - | ns |
| tAADR | 回读状态下 AE 到 AE 延迟时间 | 80 | - | ns |
| tTR | NVSTR 上升沿后 TBIT 上升沿时间 | - | 100 | ns |

| 符号 | 描述 | 规格 | | 单位 |
|---------|--------------------------------|-----|-----|----|
| | | 最小值 | 最大值 | |
| tTF | 写操作下 NVSTR 上升沿到 TBIT 下降沿的时间 | - | 30 | us |
| tTF | 页擦除操作下 NVSTR 上升沿到 TBIT 下降沿的时间 | - | 2 | ms |
| tTF | 模块擦除操作下 NVSTR 上升沿到 TBIT 下降沿的时间 | - | 10 | ms |
| tNVSTRH | NVSTR 上升沿到 AE 上升沿保持时间 | 10 | - | ns |
| tNVSTRL | NVSTR 下降沿到 TBIT 下降沿保持时间 | 50 | - | ns |
| tCS | CS 建立时间 | 10 | - | ns |
| tRCH | 读操作下 CS 保持时间 | 0 | - | ns |
| tWCH | 写操作下 CS 保持时间 | 10 | - | ns |
| tECH | 擦除操作下 CS 保持时间 | 10 | - | ns |
| tDOH | AE 有效到数据输出的时间 | 5 | - | ns |
| tOS | 读使能建立时间 | 1 | - | ns |
| tOH | 读使能保持时间 | 30 | - | ns |
| tOHR | 回读状态下读使能保持时间 | 80 | - | ns |

时序图

图 3-15 读操作模式

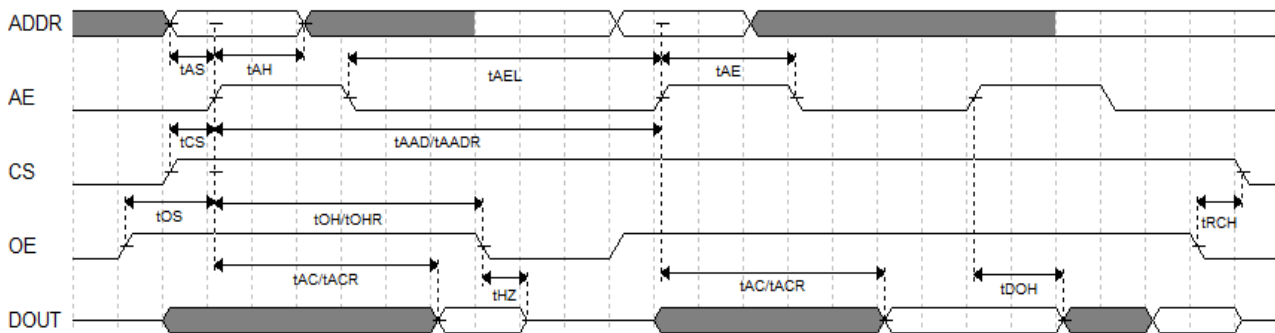


图 3-16 写操作模式

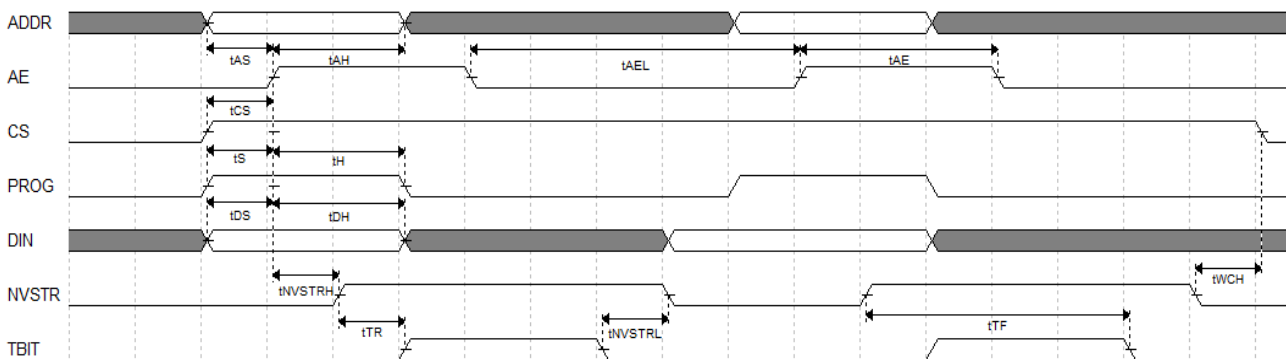


图 3-17 页擦除模式

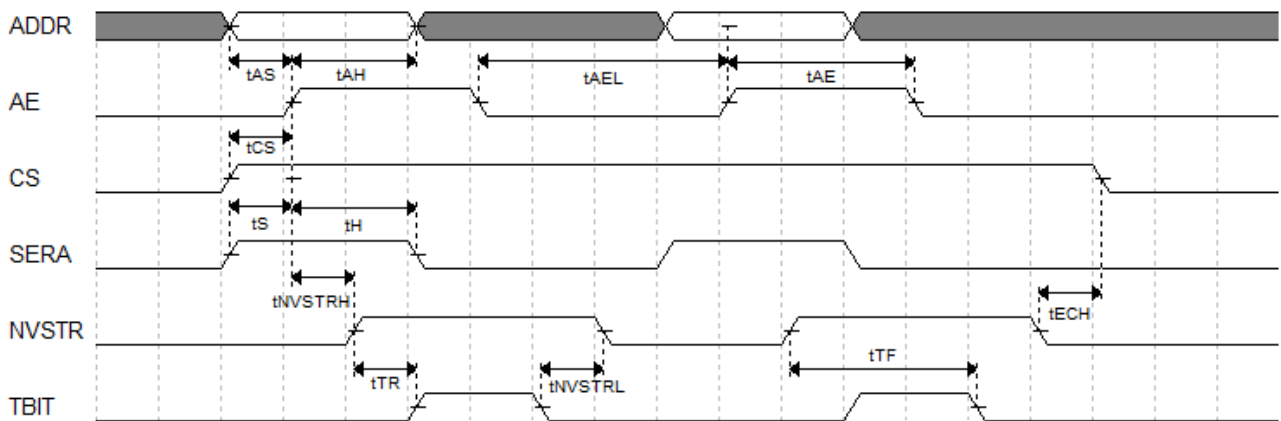
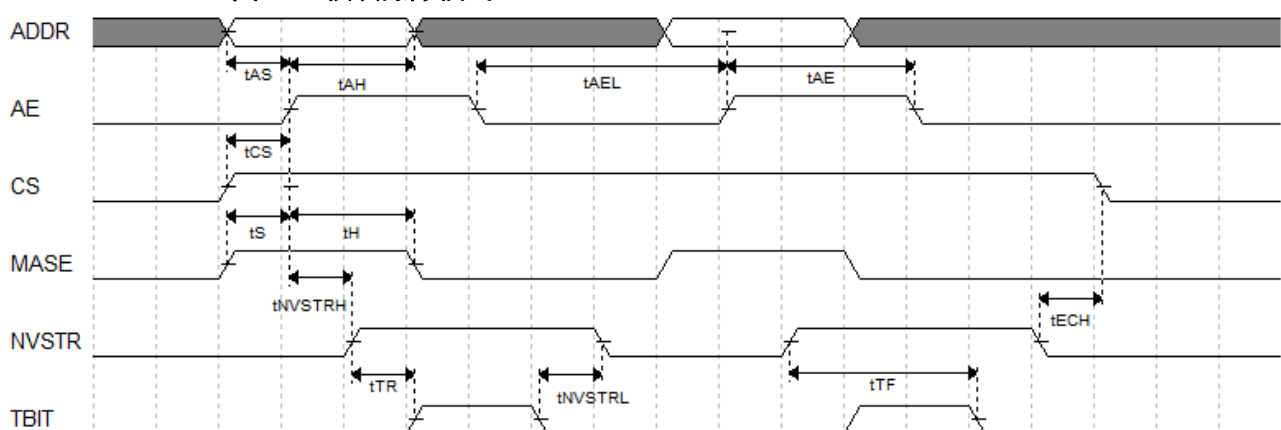


图 3-18 模块擦除模式



3.5 FLASH256K

原语介绍

FLASH256K（256Kbit User Flash）的存储空间为 256K bit。寄存器的宽度和深度是固定的，不可对其进行配置。具有非易失性和断电保存功能，但不支持初始值功能。

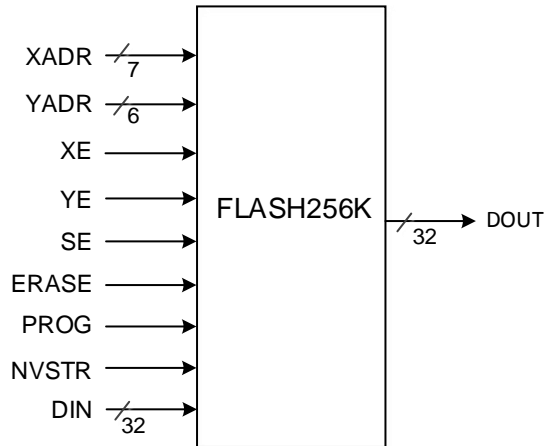
FLASH256K 由行存储单元和列存储单元组成，一行由 64 个列存储单元组成，列存储单元的容量为 32bits，行存储单元的容量为 $64 \times 32 = 2048$ bits。擦除操作支持页擦除，一页的容量为 2048 字节，即一页包含 8 行，特性如下所示：

- 10,000 次写寿命周期
- 超过 10 年的数据保存能力(+85°C)
- 数据位宽：32
- 容量：128 行*64 列*32 = 256kbits
- 页擦除能力：2,048 字节
- 快速页擦除/读写操作
- 时钟频率：40MHz

- 字写操作时间：≤16μs
- 页擦除时间：≤120ms
- 电流
 - 读电流/持续时间：2.19mA/25ns (V_{CC}) & 0.5mA/25ns (V_{CCX})(MAX)
 - 写操作/擦除操作：12/12mA(MAX)

端口示意图

图 3-19 FLASH256K 端口示意图



端口介绍

表 3-18 端口介绍

| 端口 | I/O | 描述 |
|------------|--------|--|
| DOUT[31:0] | Output | 数据输出总线。 |
| DIN[31:0] | Input | 数据输入总线。 |
| XADR[6:0] | Input | X 地址总线，访问行地址，其中 XADR[n:3]用于选择某一页，XADR[2:0] 用于选择一页中的某一行，一页由 8 行组成，一行由 64 列组成。 |
| YADR[5:0] | Input | Y 地址总线，用于选择一行存储单元中的某一列，一行由 64 列组成。 |
| XE | Input | X 地址使能信号，当 XE 为 0 时，所有行地址均不使能。 |
| YE | Input | Y 地址使能信号，当 YE 为 0 时，所有列地址均不使能。 |
| SE | Input | 检测放大器使能信号，高电平有效。 |
| PROG | Input | 写信号，高电平有效。 |
| ERASE | Input | 擦除信号，高电平有效。 |
| NVSTR | Input | Flash 数据存储信号，高电平有效。 |

操作模式

表 3-19 用户模式真值表

| 模式 | XE | YE | SE | PROG | ERASE | NVSTR |
|-------|----|----|----|------|-------|-------|
| 读模式 | H | H | H | L | L | L |
| 写模式 | H | H | L | H | L | H |
| 页擦除模式 | H | L | L | L | H | H |

注！

“H”和“L”表示高电平和低电平。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体可参考第 4 章 IP 调用。

Verilog 例化：

```
FLASH256K flash256k_inst(
    .XADR(xadr[6:0]),
    .YADR(yadr[5:0]),
    .XE(xe),
    .YE(ye),
    .SE(se),
    .ERASE(erase),
    .PROG(prog),
    .NVSTR(nvstr),
    .DIN(din[31:0]),
    .DOUT(dout[31:0])
);
```

Vhdl 例化：

```
COMPONENT FLASH256K
PORT(
    DIN:IN std_logic_vector(31 downto 0);
    XADR:IN std_logic_vector(6 downto 0);
    YADR:IN std_logic_vector(5 downto 0);
    XE:IN std_logic;
    YE:IN std_logic;
    SE:IN std_logic;
    ERASE:IN std_logic;
    PROG:IN std_logic;
    NVSTR:IN std_logic;
```

```

        DOUT:OUT std_logic_vector(31 downto 0)
    );
END COMPONENT;
 uut: FLASH256K
    PORT MAP (
        DIN=>din,
        XADR=>xadr,
        YADR=>yadr,
        XE=>xe,
        YE=>ye,
        SE=>se,
        ERASE=>erase,
        PROG=>prog,
        NVSTR=>nvstr,
        DOUT=>dout
    );

```

时序参数

表 3-20 时序参数

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|----------------|-----|------------|-----|-----|---------|
| 访问时间 | WC1 | T_{acc} | - | 25 | ns |
| | TC | | - | 22 | ns |
| | BC | | - | 21 | ns |
| | LT | | - | 21 | ns |
| | WC | | - | 25 | ns |
| 写/擦除到数据存储建立时间 | | T_{nvs} | 5 | - | μ s |
| 数据存储保持时间 | | T_{nvh} | 5 | - | μ s |
| 数据存储保持时间(整体擦除) | | T_{nvh1} | 100 | - | μ s |
| 数据存储到写建立时间 | | T_{pgs} | 10 | - | μ s |
| 写保持时间 | | T_{pgh} | 20 | - | ns |
| 写时间 | | T_{prog} | 8 | 16 | μ s |
| 写准备时间 | | T_{wpr} | >0 | - | ns |
| 擦除保持时间 | | T_{whd} | >0 | - | ns |
| 控制信号到写/擦除建立时间 | | T_{cps} | -10 | - | ns |
| SE 到读操作建立时间 | | T_{as} | 0.1 | - | ns |
| SE 脉冲的高电平时间 | | T_{pws} | 5 | - | ns |
| 地址/数据建立时间 | | T_{ads} | 20 | - | ns |
| 地址/数据保持时间 | | T_{adh} | 20 | - | ns |

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|---------------------|-----|--------------|-----|-----|---------|
| 数据保持时间 | | T_{dh} | 0.5 | - | ns |
| 读模式地址保持时间 | WC1 | T_{ah} | 25 | - | ns |
| | TC | - | 22 | - | ns |
| | BC | - | 21 | - | ns |
| | LT | - | 21 | - | ns |
| | WC | - | 25 | - | ns |
| SE 脉冲低电平时间 | | T_{nws} | 2 | - | ns |
| 恢复时间 | | T_{rcv} | 10 | - | μs |
| 数据存储时间 | | T_{hv} | - | 6 | ms |
| 擦除时间 | | T_{erase} | 100 | 120 | ms |
| 整体擦除时间 | | T_{me} | 100 | 120 | ms |
| 掉电到待机模式的 Wake-up 时间 | | T_{wk_pd} | 7 | - | μs |
| 待机保持时间 | | T_{sbh} | 100 | - | ns |
| V_{CC} 建立时间 | | T_{ps} | 0 | - | ns |
| V_{CCX} 保持时间 | | T_{ph} | 0 | - | ns |

注!

- 这些设定值可能会改变。
- 这些数值为仿真数据，在实际器件中会有改变。
- 在信号 XADR、YADR、XE 和 YE 信号有效后， T_{acc} 的开始时间为 SE 信号的上升沿。读取的数据 DOUT 被保存直到在下一有效读操作开始。
- T_{hv} 时间为写操作开始到数据下一次擦除操作之前的累积时间，同一个地址在下次擦除之前不能被写入两次；同一个存储单元在下次擦除之前不能被写入两次。这种限制是基于安全考虑的。
- 所有的波形都有 1ns 的上升沿时间和 1ns 的下降沿时间。
- 控制信号 X、YADR、XE 和 YE 信号需要至少保持 T_{acc} 的时间， T_{acc} 从 SE 的上升沿处开始。

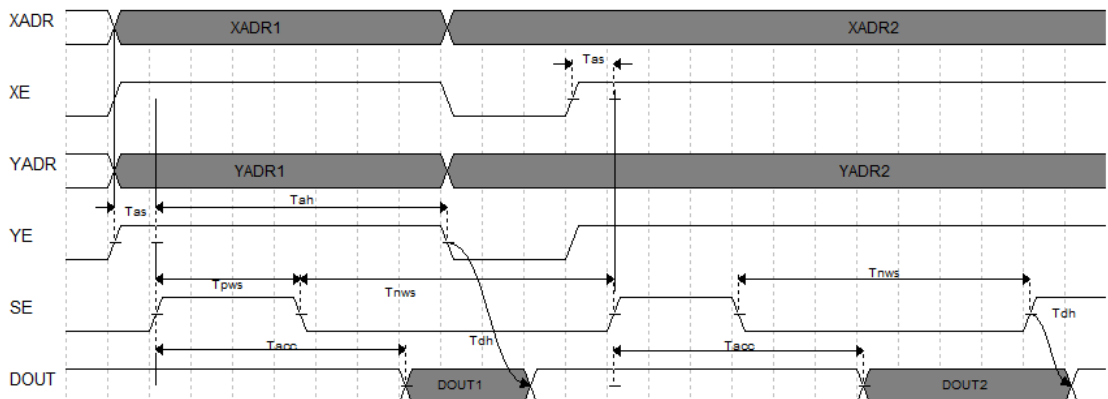
时序图**图 3-20 读操作时序**

图 3-21 写操作时序

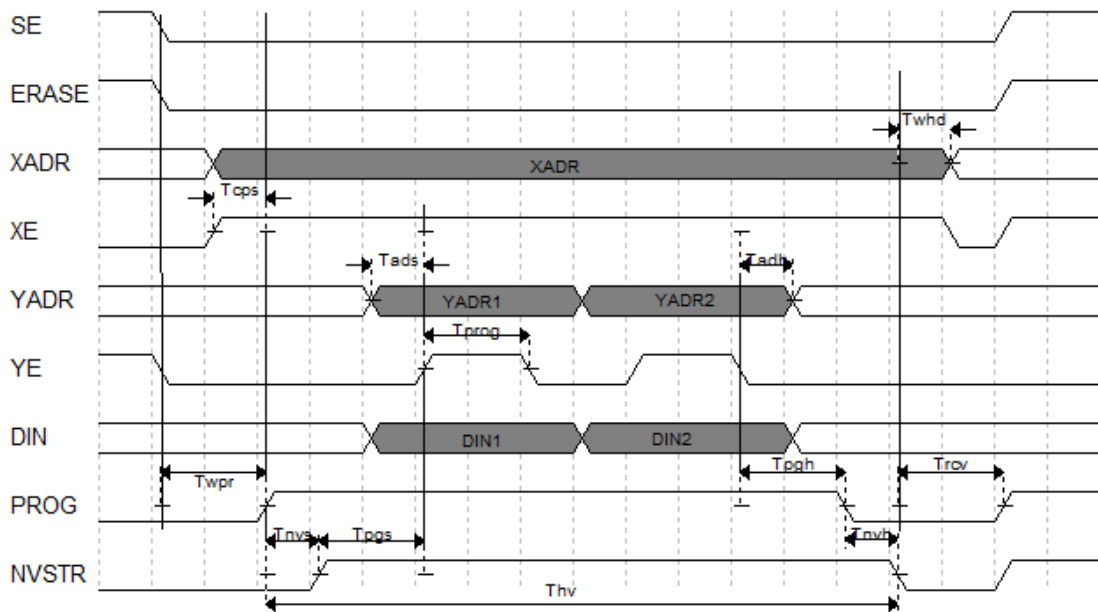
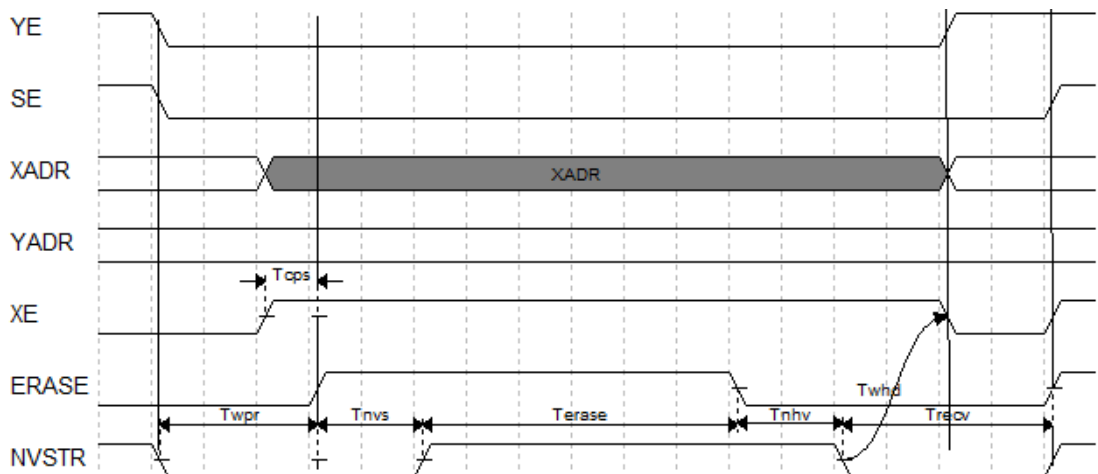


图 3-22 擦除操作时序



3.6 FLASH608K

原语介绍

FLASH608K（608Kbit Users Flash）的存储空间为 608K bit。寄存器的宽度和深度是固定的，不可对其进行配置。具有非易失性和断电保存功能，但不支持初始值功能。

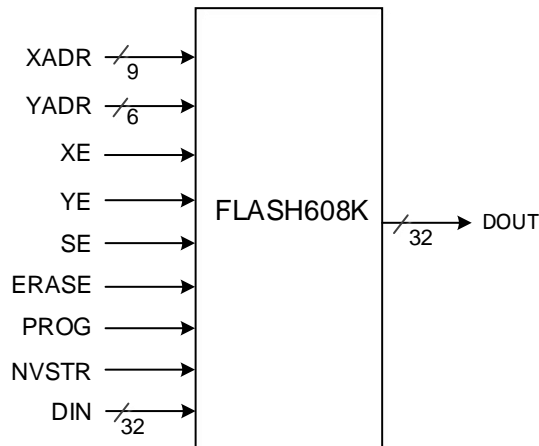
FLASH608K 由行存储和列存储单元组成，一行由 64 个列存储单元组成，列存储单元的容量为 32bits，行存储单元的容量为 $64 \times 32 = 2048$ bits。擦除操作支持页擦除，一页的容量为 2048 字节，即一页包含 8 行。特性如下所示：

- 10,000 次写寿命周期
- 超过 10 年的数据保存能力(+85°C)
- 数据位宽：32

- 容量：304 行*64 列*32 = 608Kbits
- 页擦除能力：2,048 字节
- 快速页擦除/读写操作
- 时钟频率：40MHz
- 字写操作时间：≤16μs
- 页擦除时间：≤120ms
- 电流
 - 读电流/持续时间：2.19mA/25ns (V_{CC}) & 0.5mA/25ns (V_{CCX})(MAX)
 - 写操作/擦除操作：12/12mA(MAX)

端口示意图

图 3-23 FLASH608K 端口示意图



端口介绍

表 3-21 端口介绍

| 端口 | I/O | 描述 |
|------------|--------|--|
| DOUT[31:0] | Output | 数据输出总线。 |
| DIN[31:0] | Input | 数据输入总线。 |
| XADR[8:0] | Input | X 地址总线，访问行地址，其中 XADR[n:3]用于选择某一页，XADR[2:0] 用于选择一页中的某一行，一页由 8 行组成，一行由 64 列组成。 |
| YADR[5:0] | Input | Y 地址总线，用于选择一行存储单元中的某一列，一行由 64 列组成。 |
| XE | Input | X 地址使能信号，当 XE 为 0 时，所有行地址均不使能。 |
| YE | Input | Y 地址使能信号，当 YE 为 0 时，所有列地址均不使能。 |
| SE | Input | 检测放大器使能信号，高电平有效。 |
| PROG | Input | 写信号，高电平有效。 |
| ERASE | Input | 擦除信号，高电平有效。 |

| 端口 | I/O | 描述 |
|-------|-------|---------------------|
| NVSTR | Input | Flash 数据存储信号，高电平有效。 |

操作模式

表 3-22 用户模式真值表

| 模式 | XE | YE | SE | PROG | ERASE | NVSTR |
|-------|----|----|----|------|-------|-------|
| 读模式 | H | H | H | L | L | L |
| 写模式 | H | H | L | H | L | H |
| 页擦除模式 | H | L | L | L | H | H |

注！

“H”和“L”表示高电平和低电平。

原语例化

可以直接实例化原语，也可以通过 IP Core Generator 工具产生，具体可参考第 4 章 IP 调用。

Verilog 例化：

```
FLASH608K flash608k_inst(
    .XADR(xadr[8:0]),
    .YADR(yadr[5:0]),
    .XE(xe),
    .YE(ye),
    .SE(se),
    .ERASE(erase),
    .PROG(prog),
    .NVSTR(nvstr),
    .DIN(din[31:0]),
    .DOUT(dout[31:0])
);
```

Vhdl 例化：

```
COMPONENT FLASH608K
PORT(
    DIN:IN std_logic_vector(31 downto 0);
    XADR:IN std_logic_vector(8 downto 0);
    YADR:IN std_logic_vector(5 downto 0);
    XE:IN std_logic;
    YE:IN std_logic;
    SE:IN std_logic;
```



```

ERASE:IN std_logic;
PROG:IN std_logic;
NVSTR:IN std_logic;
DOUT:OUT std_logic_vector(31 downto 0)
);
END COMPONENT;
 uut: FLASH608K
  PORT MAP (
    DIN=>din,
    XADR=>xadr,
    YADR=>yadr,
    XE=>xe,
    YE=>ye,
    SE=>se,
    ERASE=>erase,
    PROG=>prog,
    NVSTR=>nvstr,
    DOUT=>dout
  );

```

时序参数

表 3-23 时序参数

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|----------------|-----|------------|-----|-----|---------|
| 访问时间 | WC1 | T_{acc} | - | 25 | ns |
| | TC | | - | 22 | ns |
| | BC | | - | 21 | ns |
| | LT | | - | 21 | ns |
| | WC | | - | 25 | ns |
| 写/擦除到数据存储建立时间 | | T_{nvs} | 5 | - | μs |
| 数据存储保持时间 | | T_{nvh} | 5 | - | μs |
| 数据存储保持时间(整体擦除) | | T_{nvh1} | 100 | - | μs |
| 数据存储到写建立时间 | | T_{pgs} | 10 | - | μs |
| 写保持时间 | | T_{pgh} | 20 | - | ns |
| 写时间 | | T_{prog} | 8 | 16 | μs |
| 写准备时间 | | T_{wpr} | >0 | - | ns |
| 擦除保持时间 | | T_{whd} | >0 | - | ns |
| 控制信号到写/擦除建立时间 | | T_{cps} | -10 | - | ns |
| SE 到读操作建立时间 | | T_{as} | 0.1 | - | ns |

| 用户模式 | 参数 | 符号 | 最小值 | 最大值 | 单位 |
|---------------------|-----|--------------|-----|-----|---------|
| SE 脉冲的高电平时间 | | T_{pws} | 5 | - | ns |
| 地址/数据建立时间 | | T_{ads} | 20 | - | ns |
| 地址/数据保持时间 | | T_{adh} | 20 | - | ns |
| 数据保持时间 | | T_{dh} | 0.5 | - | ns |
| 读模式地址保持时间 | WC1 | T_{ah} | 25 | - | ns |
| | TC | - | 22 | - | ns |
| | BC | - | 21 | - | ns |
| | LT | - | 21 | - | ns |
| | WC | - | 25 | - | ns |
| SE 脉冲低电平时间 | | T_{nws} | 2 | - | ns |
| 恢复时间 | | T_{rcv} | 10 | - | μ s |
| 数据存储时间 | | T_{hv} | - | 6 | ms |
| 擦除时间 | | T_{erase} | 100 | 120 | ms |
| 整体擦除时间 | | T_{me} | 100 | 120 | ms |
| 掉电到待机模式的 Wake-up 时间 | | T_{wk_pd} | 7 | - | μ s |
| 待机保持时间 | | T_{sbh} | 100 | - | ns |
| V_{CC} 建立时间 | | T_{ps} | 0 | - | ns |
| V_{CCX} 保持时间 | | T_{ph} | 0 | - | ns |

注!

- 这些设定值可能会改变。
- 这些数值为仿真数据，在实际器件中会有改变。
- 在信号 XADR、YADR、XE 和 YE 信号有效后， T_{acc} 的开始时间为 SE 信号的上升沿。读取的数据 DOUT 被保存直到在下一次有效的读操作开始。
- T_{hv} 时间为写操作开始到数据下一次擦除操作之前的累积时间，同一个地址在下次擦除之前不能被写入两次；同一个存储单元在下次擦除之前不能被写入两次。这种限制是基于安全考虑的。
- 所有的波形都有 1ns 的上升沿时间和 1ns 的下降沿时间。
- 控制信号 X、YADR、XE 和 YE 信号需要至少保持 T_{acc} 的时间， T_{acc} 从 SE 的上升沿处开始。

时序图

图 3-24 读操作时序

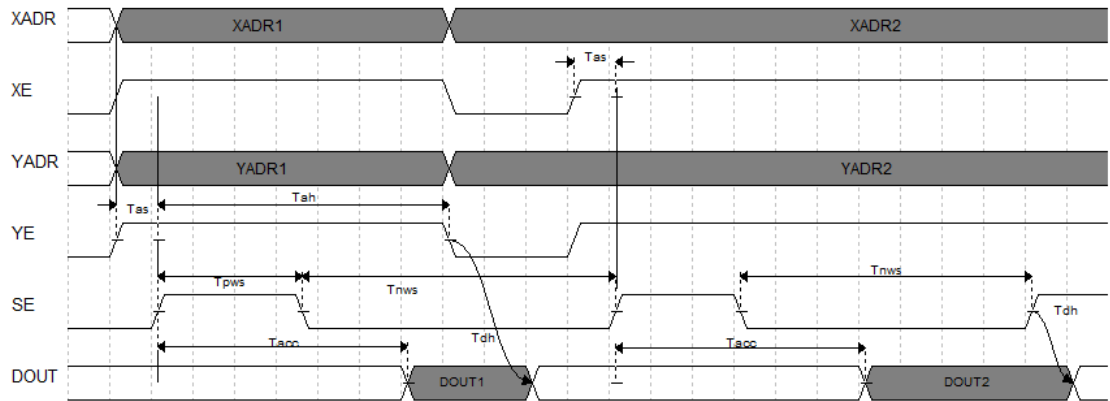


图 3-25 写操作时序

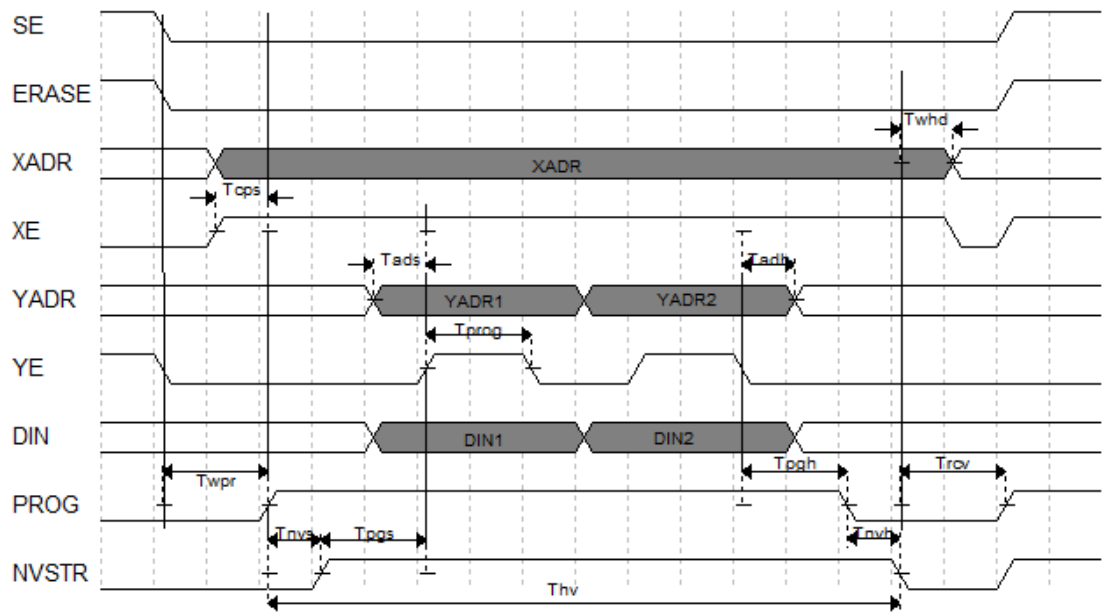
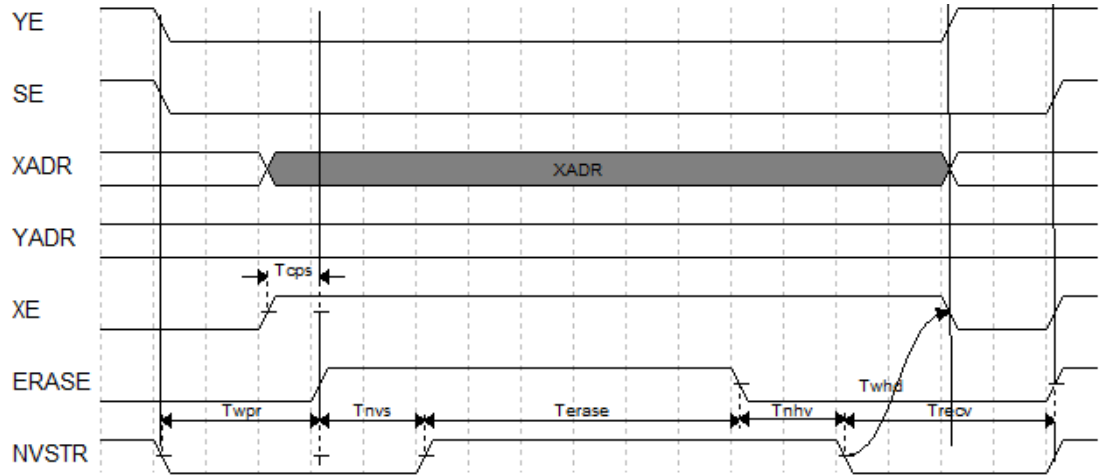


图 3-26 擦除操作时序



4 IP 调用

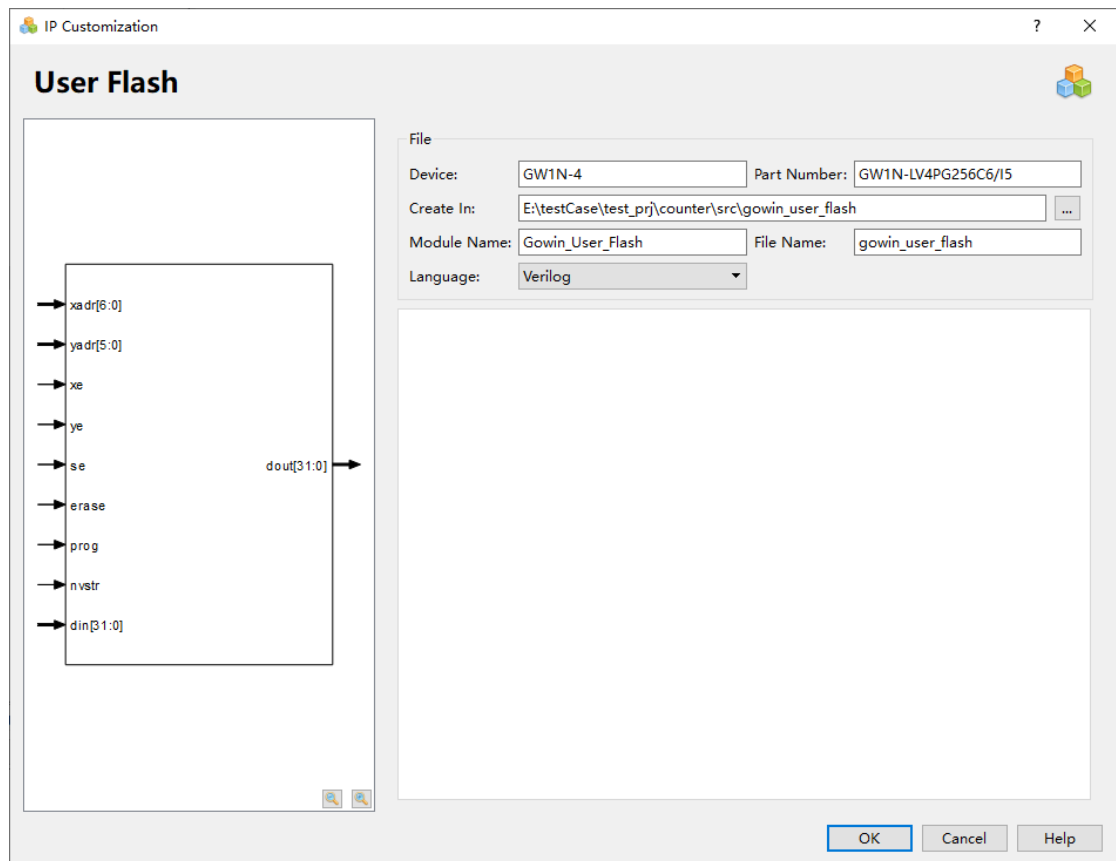
IP Core Generator 可以产生硬核 User Flash 和软核 Gowin Flash Controller IP，Gowin Flash Controller IP 的功能及 GUI 调用等，参考 [IPUG901](#)，Gowin Flash Controller IP 用户指南。硬核 User Flash 可通过 IP Core Generator 界面 Hard Module 中的“User Flash”产生。

在 IP Core Generator 界面中单击“User Flash”，界面右侧会显示 User Flash 的相关信息概要。

IP 配置

在 IP Core Generator 界面中，双击“User Flash”，弹出 User Flash 的“IP Customization”窗口，该窗口包括“File”配置框和端口显示框图以及“Help”按钮，如图 4-1 所示。

图 4-1 User Flash 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生的 IP 设计文件的相关信息。

- **Device:** 显示已配置的 Device 信息；
- **Part Number:** 显示已配置的 Part Number 信息；
- **Language:** 配置产生的 IP 设计文件的硬件描述语言。选择右侧下拉列表框，选择目标语言，支持 Verilog 和 VHDL；
- **Module Name:** 配置产生的 IP 设计文件的 module name。在右侧文本框可重新编辑模块名称。Module Name 不能与原语名称相同，若相同，则报出 Error 提示；
- **File Name:** 配置产生的 IP 设计文件的文件名。在右侧文本框可重新编辑文件名称；
- **Create In:** 配置产生的 IP 设计文件的目标路径。可在右侧文本框中重新编辑目标路径，也可通过文本框右侧选择按钮选择目标路径。

2. 端口显示框图

User Flash 的输入位宽与 Device 的选择有关，根据不同的 Device 信息会在端口显示框图中显示当前 IP Core 的配置结果示例框图，如图 4-1 所示。

3. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面。Help 页面包括 IP Core

的概要介绍。

IP 生成文件

IP 窗口配置完成后，产生以配置文件“File Name”命名的三个文件，以默认配置为例进行介绍：

- IP 设计文件“gowin_user_flash.v”为完整的 verilog 模块，根据用户的 IP 配置，产生实例化的 User Flash；
- IP 设计使用模板文件 gowin_user_flash_tmp.v，为用户提供 IP 设计使用模板文件；
- IP 配置文件：“gowin_user_flash.ipc”，用户可加载该文件对 IP 进行配置。

注！

如配置中选择语言是 VHDL，则产生的前两个文件名后缀为.vhd。

