**SPI通信**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Version Control | | | |
| Version | Date | Author | Changes |
| 1.0.0 | 2024.4.13 | 王雨霄 | 实现了标准4线SPI通信协议 |
| 1.0.1 | 2024.7.26 | 王雨霄 | 增添了3线SPI模式 |
| 1.0.2 | 2024.7.26 | 王雨霄 | 使用IO原语驱动3线SPI |

1. **功能概述**

本设计对 3线/4线标准SPI通信协议进行了 Verilog 实现，主要功能特性如下：

1. 内置分频器，能够在自定义系统时钟频率下进行任意速率的SPI通信（SCK经系统时钟计数分频得到，要求系统时钟频率为SPI速率的4倍及以上）；
2. 支持只写、只读（通常用于传感器数据读取）、写+读（通过写入指令来控制读取某一寄存器的值，是最常用的寄存器读取方式）三种工作方式；
3. 支持动态切换SPI时钟极性（CPOL）、时钟相位（CPHA）参数；
4. 支持动态设置SPI本次读/写传输数据的位宽；
5. 支持与 AXIS 接口进行连接；
6. **接口概述**

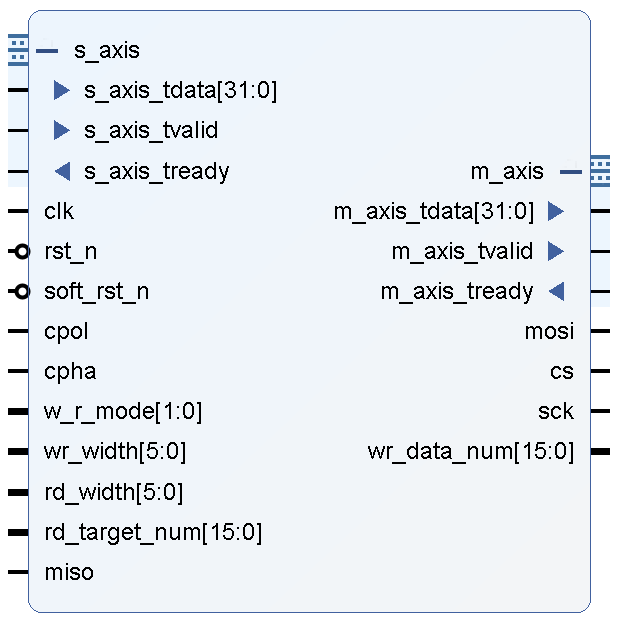
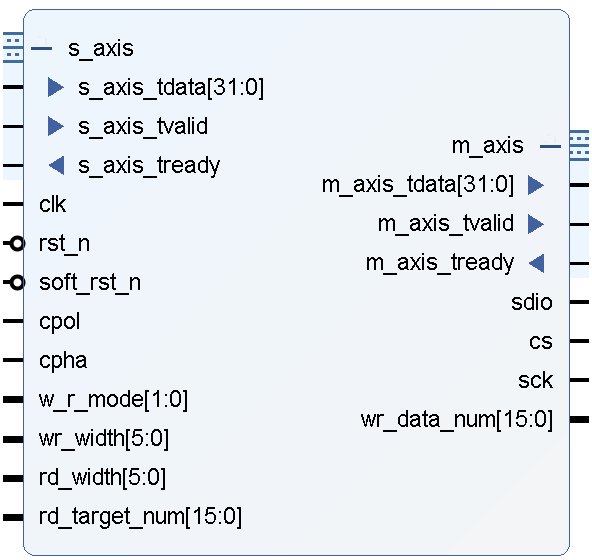
** **

图1. 模块接口



图2. 接口功能表

1. **控制接口详述**

本节对图2橙色部分所示的IP控制接口的功能及配置方式进行详细叙述。

cpol接口及cpha接口用于设置sck的时钟极性及时钟相位，二者共同决定SPI主从机在接收数据时的采样行为：

|  |  |  |
| --- | --- | --- |
| 时钟极性CPOL | 时钟相位CPHA | 采样时序 |
| 0（SCK空闲为低） | 0（在奇数个时钟边沿采样） |  |
| 0（SCK空闲为低） | 1（在偶数个时钟边沿采样） |  |
| 1（SCK空闲为高） | 0（在奇数个时钟边沿采样） |  |
| 1（SCK空闲为高） | 1（在偶数个时钟边沿采样） |  |

图3. 采样时序

w\_r\_mode接口用于设置SPI工作模式：

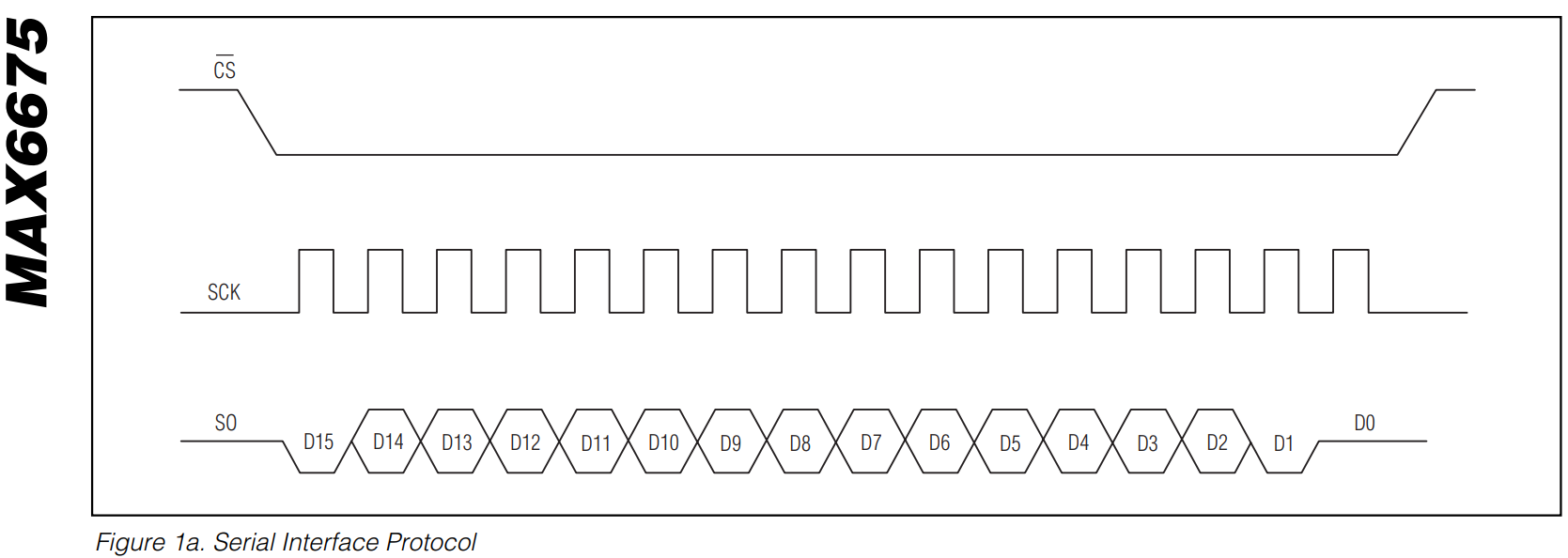


图4. MAX6675读时序

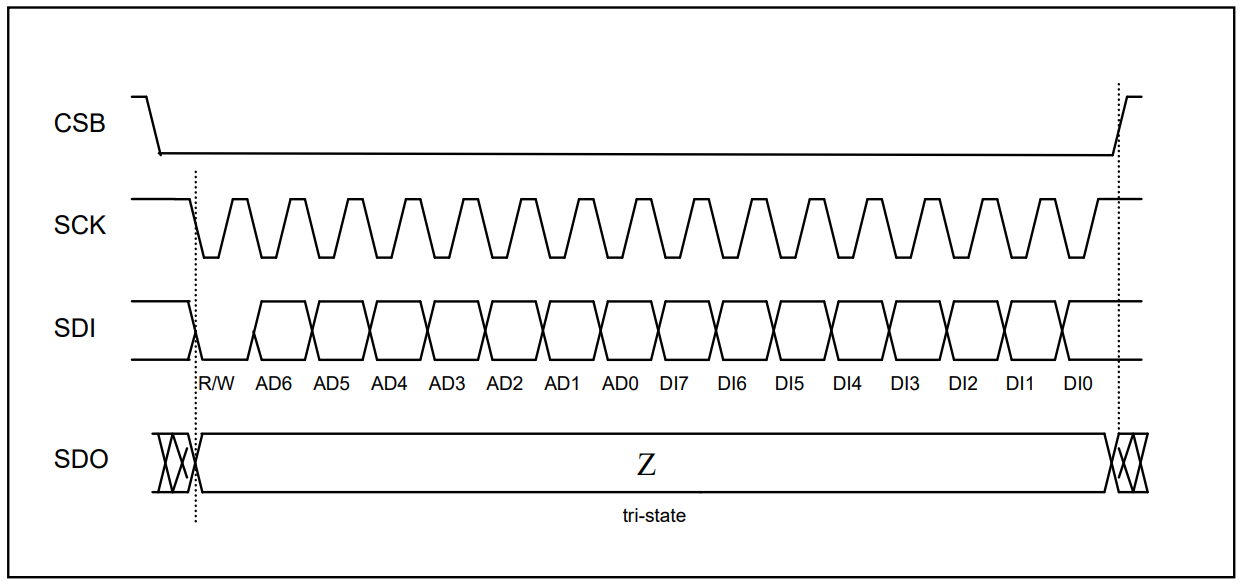


图5. BMI160写时序

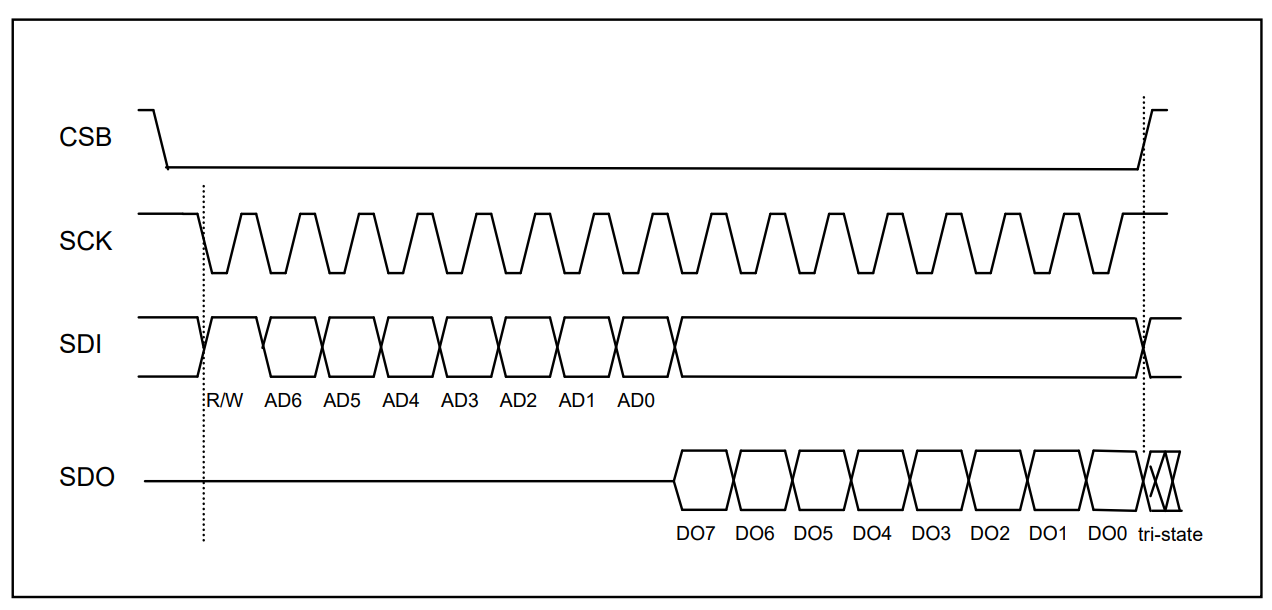


图6. BMI160读时序

对于如上图4所示的MAX6675温度传感器读操作，其时序较为简单，传感器上电后自动向MISO接口发送温度数据而无需写入任何指令，主机仅需从MISO接口进行数据读取，对于此类情况，应将w\_r\_mode设为2’b00-只读模式；对于如上图5所示的BMI160陀螺仪传感器写操作，主机仅需向MOSI接口进行数据写入，此时应将w\_r\_mode设为2’b01-只写模式；对于如上图6所示的BMI160陀螺仪传感器读操作，其时序较为复杂，主机需先向MOSI接口写入“控制位+目标寄存器地址”，再等待传感器从MISO接口返回数据，此时的读操作是在主机指令控制下的读取，应将w\_r\_mode设为2’b10-写+读模式。

wr\_width接口和rd\_width接口用于设置SPI传输时读写数据的比特位宽——对于如上图4所示的只读操作，wr\_width=0，rd\_width=16；对于如上图5所示的只写操作，wr\_width=16，rd\_width=0；对于如上图6所示的写+读操作，wr\_width=8，rd\_width=8。

1. **参数配置**

**1、system\_clk**

设定系统时钟频率（单位：Hz）；例如系统时钟为50MHz时，该参数设为50000000。

**2、spi\_rate**

设定sck时钟频率（单位：Hz）；要求system\_clk/spi\_rate≥4，否则默认按照system\_clk/4作为sck时钟频率。

**3、cs\_buff\_time**

cs在两次低电平之间保持高电平的缓冲时间（单位：一个数据长度）。

**4、sck\_buff\_time**

cs拉低-sck启动/sck停止-cs拉高之间的缓冲时间（单位：一个数据长度）。

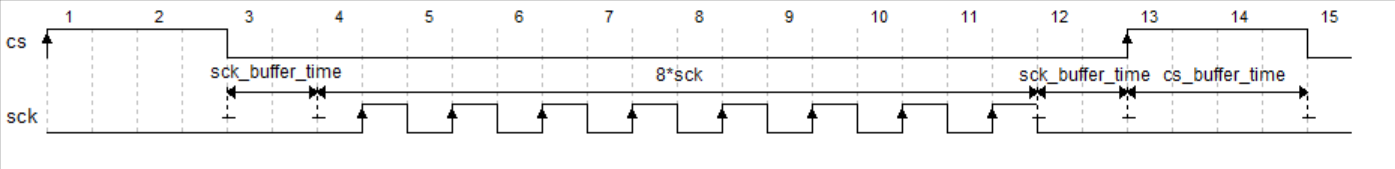


图7. cs\_buffer\_time/sck\_buffer\_time时序

**5、mode**

设置SPI工作在3线模式或4线模式。

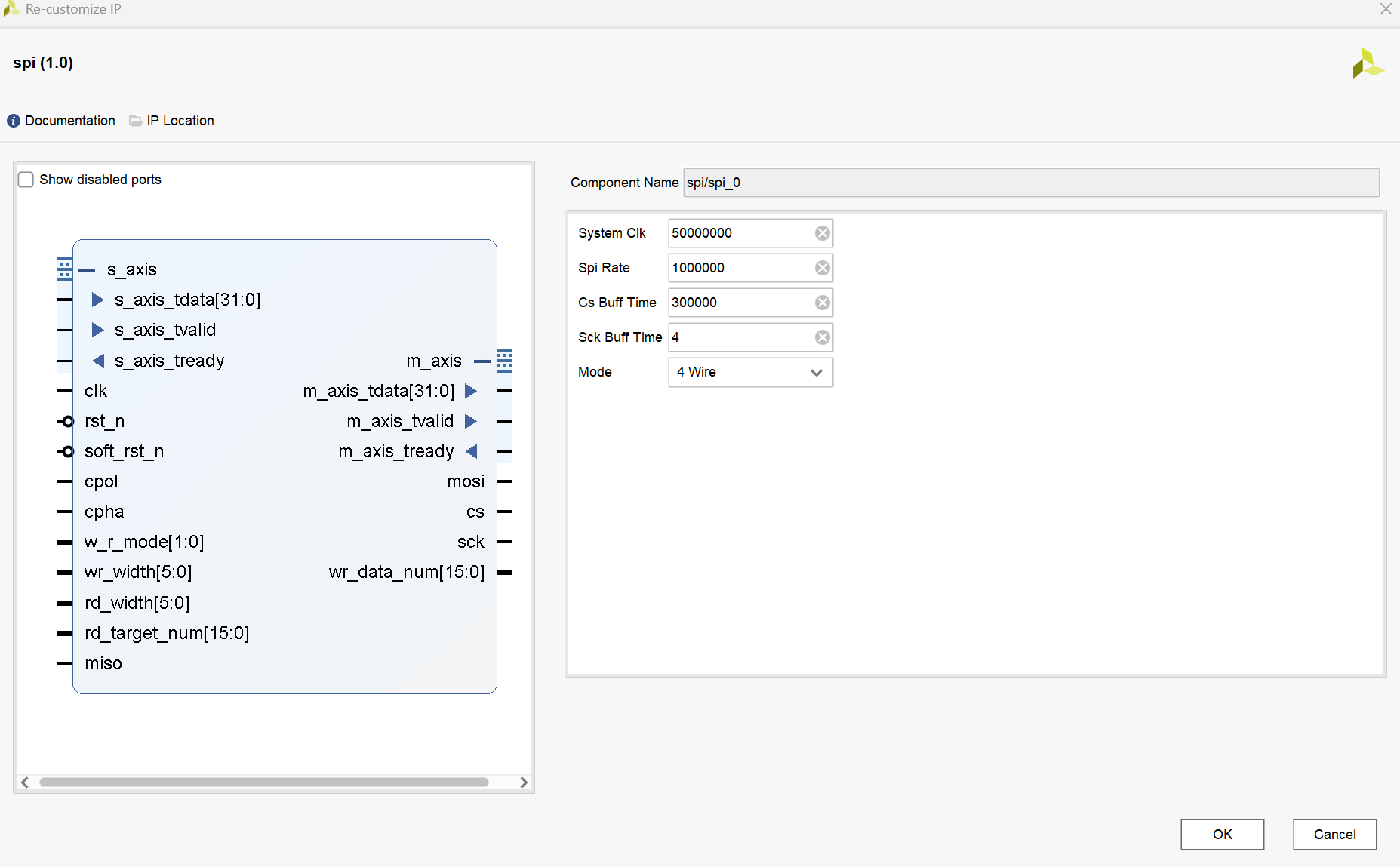
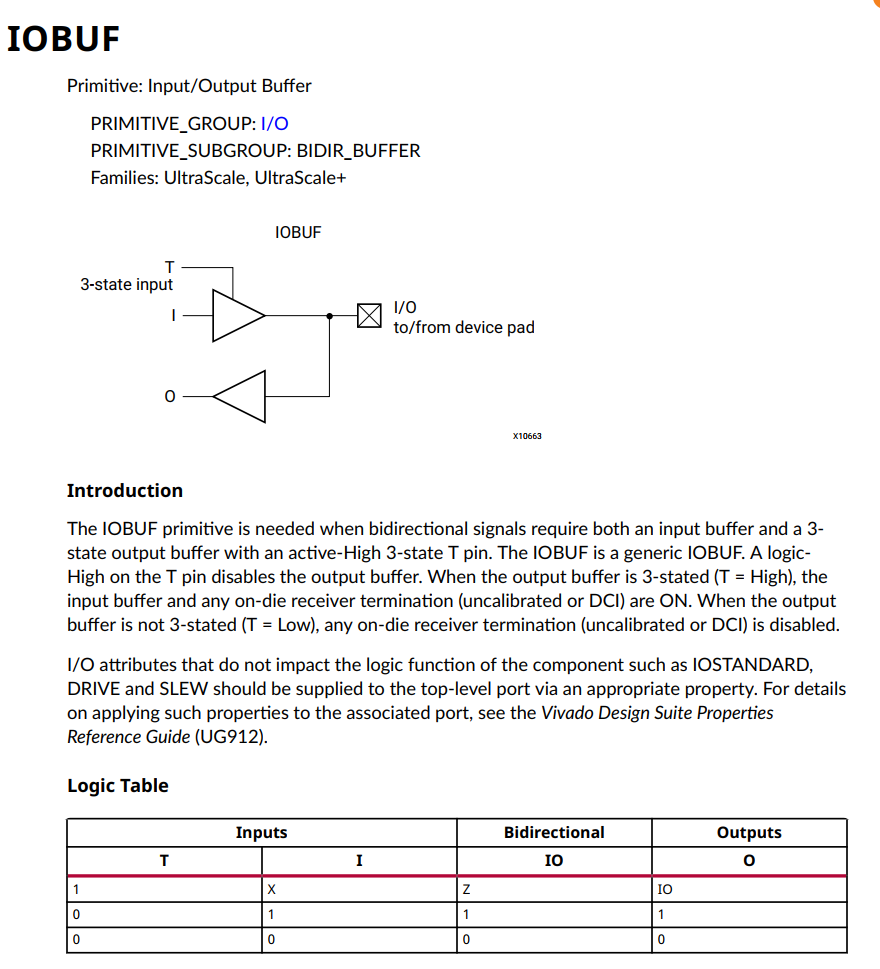


图8. 将设计导出为IP核后的参数配置界面

**附录：**



上图是XILINX器件中IOBUF的结构，其中的I、O端口是IOBUF的输入、输出；当T=1时，IOBUF的信号流向为IO➡O（外部设备➡IOBUF➡FPGA，FPGA接受输入）；当T=0时，IOBUF的信号流向为I➡IO（FPGA➡IOBUF➡外部设备，FPGA对外输出）。