# AXI-LITE设计文档

## 基本介绍

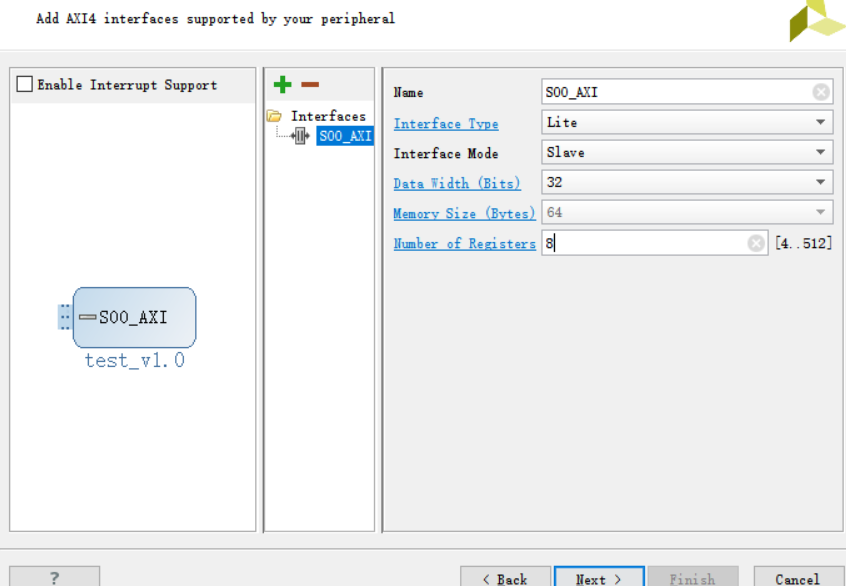
### 简介

**AXI-LITE SLAVE模块:**

**[实现功能]**主机通过PCIE以AXI\_Lite 总线写和读AXI\_Lite Slave 模块的内部寄存器 Slave寄存器的值来传递命令，如改变用户逻辑发送数据的地址等参数。在Xilinx自动生成的AXI4 interface的基础上进行功能扩充。

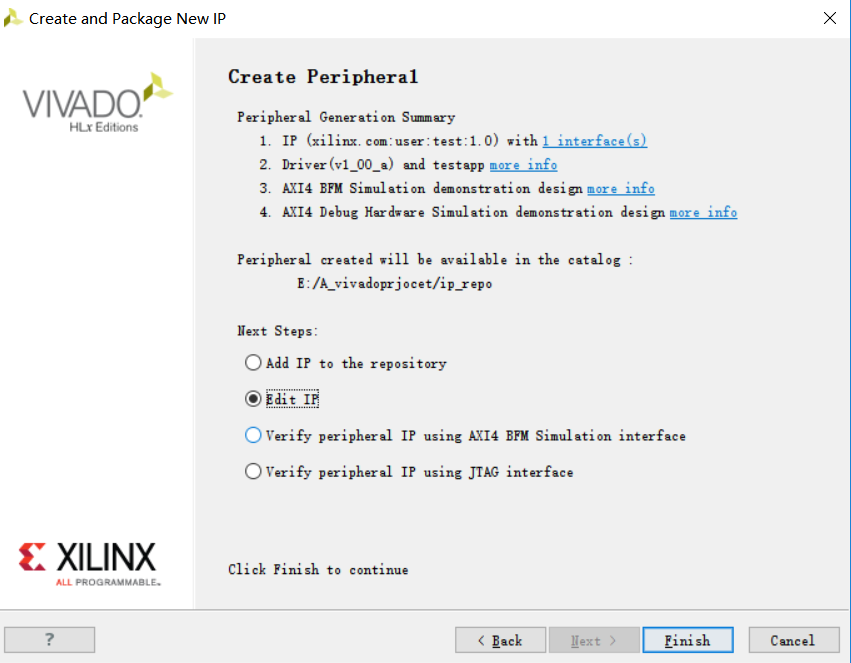
**[操作步骤]**

1. 用户可以通过”Tool—Creat and Package New Ip—Creat a New AXI Peripheral”来选择产生Slave Ip的interface ,下图为Slave 模块基本参数设置.

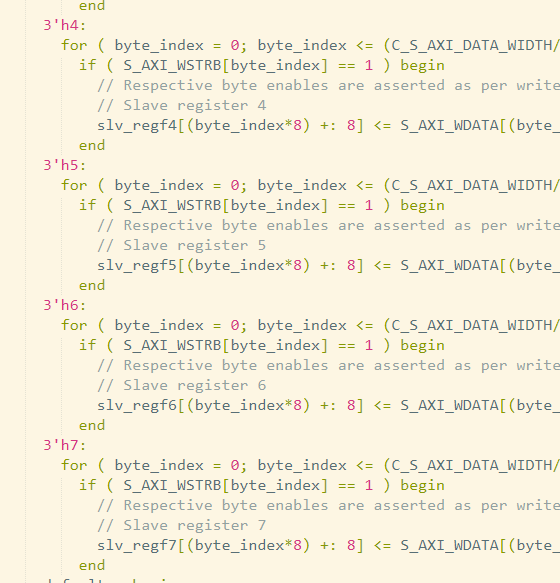
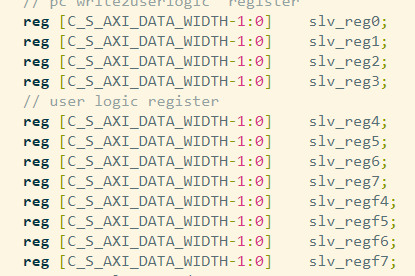


选择产生8组寄存器,4组用来写,4组用来读.

1. 完成基本配置后,点击Edit IP 来修改Slave 模块源代码,用以增加额外发送数据功能.

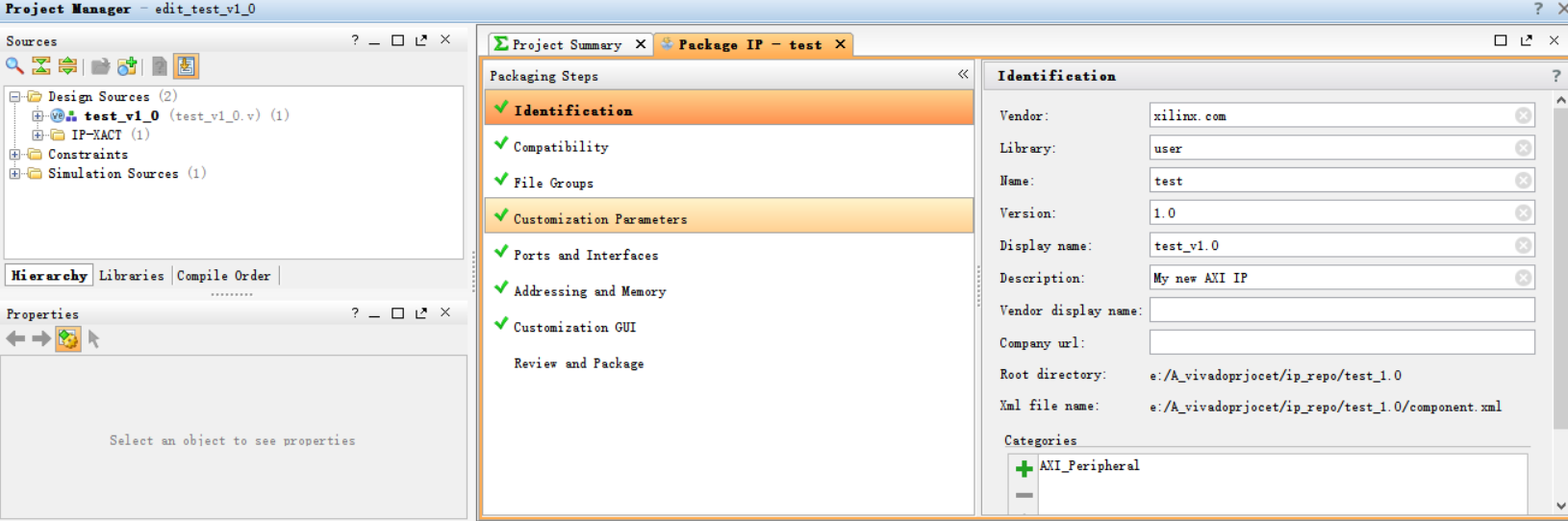


1. 在这个自动生成的IP基础上,可以自定义一些功能,例如规定:PC只写前四个寄存器(reg0—reg3),后面四个寄存器(reg4—reg7)用来存储发送数据过程中生成的信息(比如发送的数据量),故不能去执行PC写操作. PC读操作则是8个寄存器都可以读.这样就要在原来代码的基础上进行修改.关键代码如下:



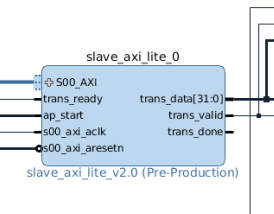
定义四个多余的寄存器,写地址为后四个寄存器的时候,即是写入这四个多余的寄存器中.

1. 代码修改完成后需要进行如下图的”重新封装”,每一个步骤都完成且无报错后点击”Repackage Ip”



5.**建议**:每一个IP都去建一个专门的Project去管理,而不是在所作工程上去添加. 当工程需要这个IP的时候,只需在现工程的” Tools—Project Settings “里面去添加IP的地址就可以完成添加IP操作.这样更加方便IP源代码的修改和Repackage IP的过程

**[模块框图]**



SLAVE模块的框图:

**[输入输出]**

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | | 输出 | |
| 信号 | 说明 | 信号 | 说明 |
| Trans\_ready | Master接受数据ready信号 | Trans\_data | 发送给Master的数据 |
| S00\_AXI | AXI总线 | Trans\_valid | 发送给Master有效信号 |
| ap\_start | 发送数据开始信号 | trans\_done | 发送Master数据完成信号 |
| S00\_axi\_aclk | 用户时钟(100M) |  |  |
| S00\_axi\_aresetn | 全局复位信号 |  |  |

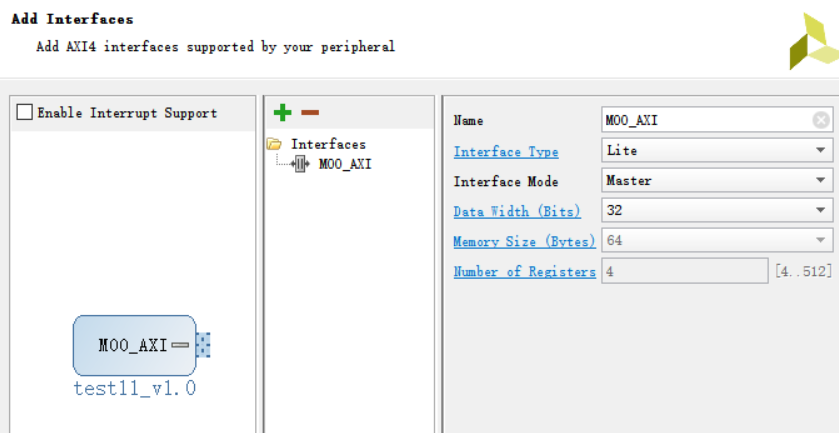
**AXI-LITE MASTER模块:**

**[实现功能]** Axi\_Lite Master模块是将从Slave模块收到的数据以AXI\_LITE的方式去读写DDR.原始IP内部定义了起始地址和起始数据,在数据宽度为32位的情况下,每次发送给DDR的地址自动+4,每次发送的值自动+1.

经过在Xilinx自动生成的AXI4 interface的基础上进行功能扩充，,MASTER可以发送一组用户定义(如斐波那契数列)数据到Slave模块传来的地址中去.

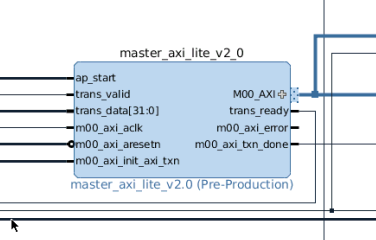
**[操作步骤]**

1. 用户可以通过”Tool—Creat and Package New Ip—Creat a New AXI Peripheral” 来选择产生Master IP,同时可以选择IP的具体参数.



1. 接下来的步骤同Slave 模块操作,只是需要改变不同的代码再封装.

**[模块框图]**

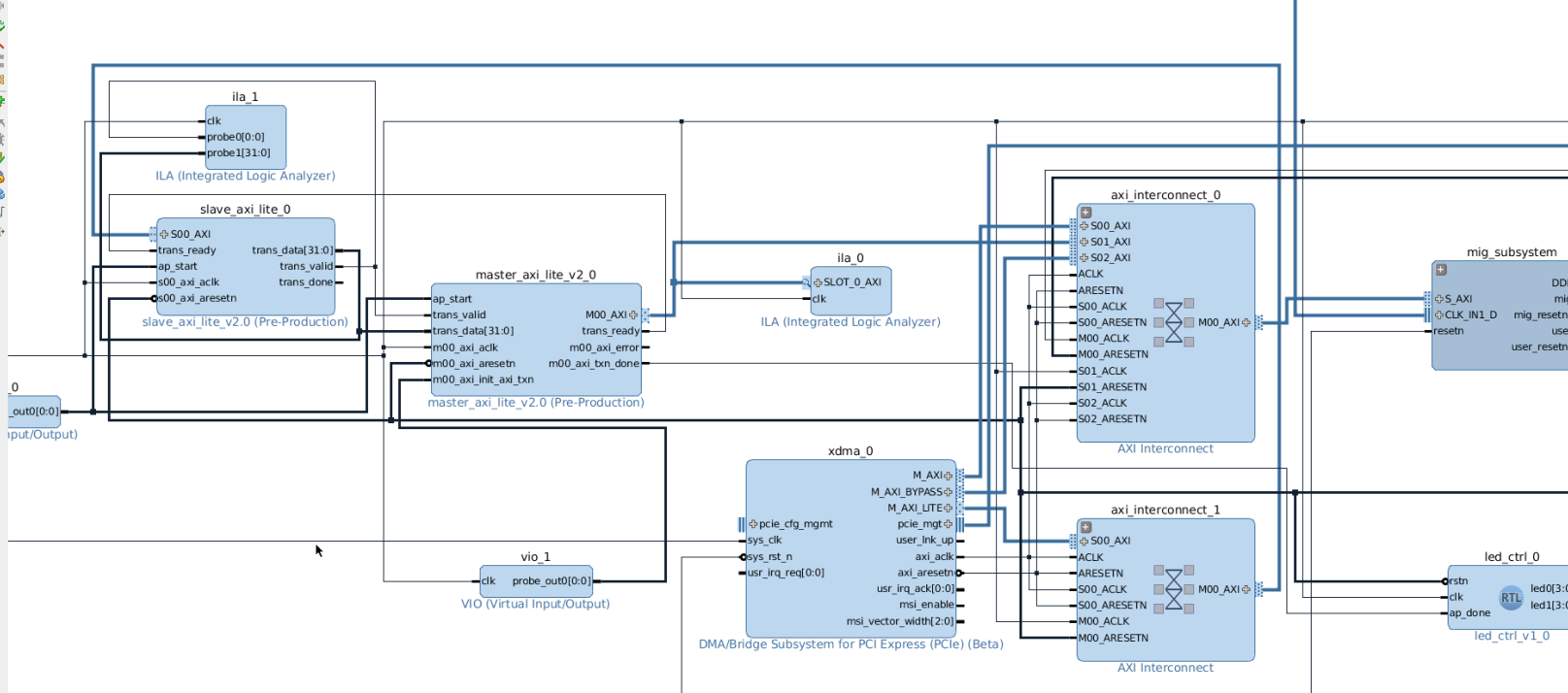


Master 模块框图

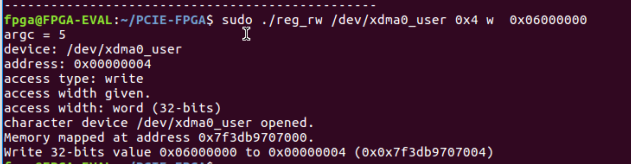
**[输入输出]**

|  |  |  |  |
| --- | --- | --- | --- |
| 输入 | | 输出 | |
| 信号 | 说明 | 信号 | 说明 |
| Trans\_data | Slave发送给Master的数据 | M00\_axi\_txn\_done | 发送数据完成信号 |
| Trans\_valid | 发送到Master数据有效信号 | Trans\_ready | Master接受数据ready信号 |
| ap\_start | 通过ap\_start 来拉高trans\_ready | trans\_done | 发送Master数据完成信号 |
| m00\_axi\_aclk | 用户时钟(100M) | M00\_AXI | AXI总线 发送数据给DDR |
| m00\_axi\_aresetn | 全局复位信号 |  |  |
| M00\_axi\_init\_axi\_txn | Master发送数据开始信号 |  |  |

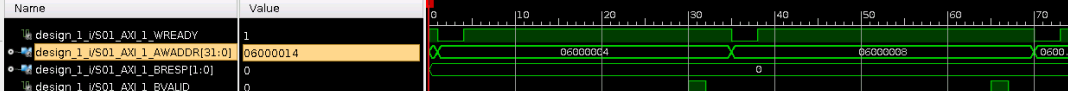
**[系统框图]**



**[操作步骤]:**



1. 通过PCIE写给Slave模块的”slave\_reg1”寄存器的地址0x0600\_0000;
2. 通过VIO-0连接”AP\_START”端口,置1,启动Slave模块传输数据给MASTER模块.
3. 通过VIO-1连接”M00\_axi\_init\_axi\_txn”启动Master传输数据给DDR
4. 通过ILA观察Master的输出端M00\_Axi,结果如下:

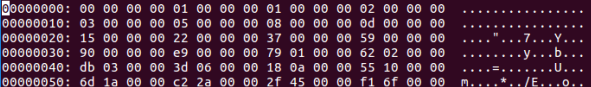


发送地址从0x0600\_0000以4位单位递增



发送数据为斐波那契数列

1. 查询DDR的0x06000000地址数据:



验证数据正确的传输到DDR的对应地址中.

柯乔

Revised by Steven NI

2017.10.12