

实验四 Vivado中IP核的应用

实验目的：

- 理解 IP 核的用途，初步掌握 IP 核的使用方法
- 学会利用 IP 核设计数字电路
- 通过与手写模块的对比，初步理解使用 IP 核的优点

实验原理：

1. IP 简介：

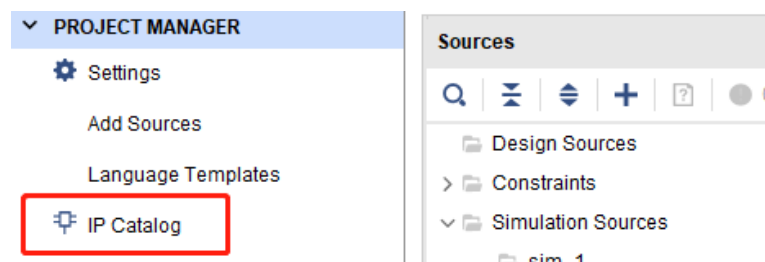
IP，全称 intellectual property。以 IP 为中心的设计流程（IP-Centric Design Flow）是 Vivado 提供的一个功能，允许用户将来自各种设计的 IP 模块添加到自己的设计中，就如同软件开发中调用其他人编写的库一样。

该功能的核心是一个具有拓展性的 IP Catalog（IP 目录），其中就包含本实验使用到的 Clock Wizard 等由赛灵思 Xilinx 提供的即插即用 IP（Plug-and-Play IP）。IP Catalog 可通过多种方式进行拓展，包括借助 MATLAB 中 Simulink 算法设计的模块、借助 C/C++ 算法设计的模块、使用 Vivado IP 封装器工具封装的模块等。

本实验主要涉及即插即用 IP 的使用，对拓展 IP 的方法不做过多介绍。对 IP 感兴趣的同学，可以前往 AMD 首页(<https://www.amd.com/zh-cn.html>)搜索 UG939 及 UG896 进行深入学习。

2. IP 核调用基本操作：

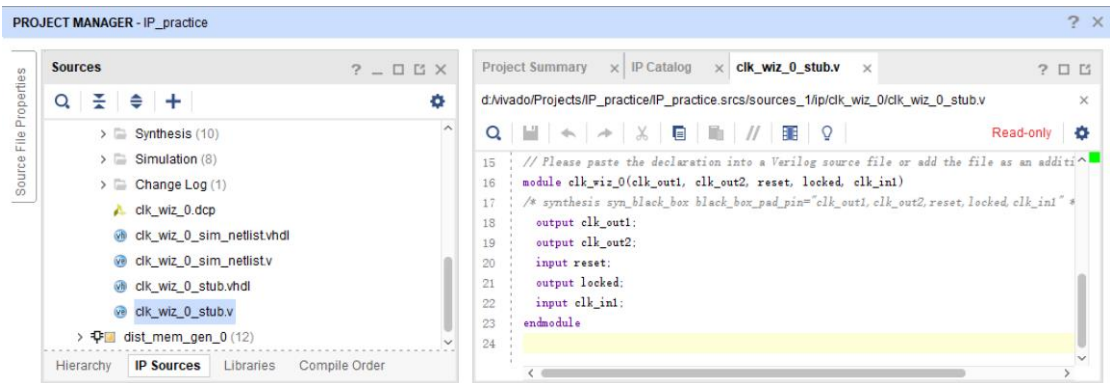
可以在 Vivado 的 IP Catalog 中选取想使用的 IP 核。IP Catalog 的位置如图所示：



找到想使用的 IP 核后，双击对其进行配置（Customize）。配置完成后，可以选择 IP 核的综合方式：global/out-of-context。选择前者将使 IP 核与项目其他部

分一起被综合，而后者将对 IP 核进行单独综合，使其独立于其他部分的综合过程。

完成后，即可在 Sources 的 IP Sources 选项卡中看到其相关文件。打开名字如 xxx_stub.v 的文件，可查看该 IP 核的模块声明，然后就可以仿照该声明在代码中调用 IP 核了：



3. 使用 IP 核构造 PLL（锁相环）基本操作：

PLL（Phase Locked Loop，锁相环），能够对输入的周期信号进行任意分频、倍频、相位调整、占空比调整，从而输出期望时钟。也可以对不进行分频倍频的时钟信号进行优化，使输出的时钟信号比输入的时钟信号在抖动方面性能更好。

构造 PLL 所用到的 IP 核是位于 FPGA Features and Design-Clocking 目录下的 Clocking Wizard，双击以进行配置。基本配置选项如下：

Clocking Options: ①Primitive 中选择 PLL；②Input Clock Information 为输入时钟的配置，可以保持不动（100.00MHz，恰好是系统时钟频率）。

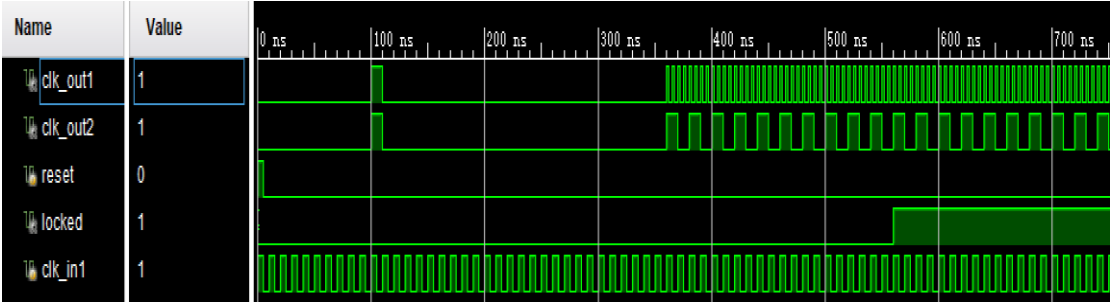
Output Clocks: 可如图设置输出时钟，此例中第一个为 2 倍频（200MHz），第二个为 2 分频（50MHz）。

Clocking Options		Output Clocks		Port Renaming		PLLE2 Settings		Summary	
The phase is calculated relative to the active input clock.									
Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)					
		Requested	Actual	Requested	Actual				
<input checked="" type="checkbox"/> clk_out1	clk_out1	200.000	200.000	0.000	0.000				
<input checked="" type="checkbox"/> clk_out2	clk_out2	50.000	50.000	0.000	0.000				

配置完成，点击 OK 后选择想要的综合方式后，完成 IP 核创建。如果在配置时没有修改 Component Name，则 IP 核名称应为 clk_wiz_0。此时，即可参照

clk_wiz_0_stub.v，在项目其他部分中例化 IP 核以进行使用。示例代码见 PLLtop_example.v。

为观察 PLL 工作现象，编写 testbench 文件，详见 PLLsim_example.v。仿真结果如下：



可见，复位 PLL 后，需要一定的准备时间，准备完成后 locked 信号变为高电平，PLL 开始正常工作，clk_out1、clk_out2 的频率与配置 IP 核时一致。

4. 使用 IP 核构造 RAM（随机访问存储器）基本操作（选做）：

RAM（Random Access Memory，随机访问存储器），可以对任意一个指定的地址写入或读出数据，分为单端口 RAM（Single Port RAM，读写数据都用一个地址端口，读写操作不能同时进行）、简双端口 RAM（Single Dual Port RAM，一个端口只负责读，另一个端口只负责写）、真双端口 RAM（True Dual Port RAM，两个端口都可以进行数据读写）等。

以单端口 RAM 为例，构造所用到的 IP 核是位于 RAMs & ROMs & BRAM 目录下的 Block Memory Generator，双击以进行配置。基本配置选项如下：

Basic: Memory Type 选择 Single Port RAM，其他可保持默认。
Port A Options: ①Memory Size 中 Width 为数据宽度，为 RAM 中最小存储单元的大小，此例设置为 32；Depth 为数据深度，为 RAM 中最小存储单元的数目，此例设置为 256，则 RAM 的容量为 256×32bits；Operating Mode 此例选择 No Change（不同操作模式之间的区别体现在读取数据的行为上，可见文末的图解）；Enable Port Type 选择“Use ENA Pin”，启用使能信号接口。②Port A Optional Output Registers 中，取消所有勾选（每勾选一个寄存器，读取数据的总延迟就会增加一个周期，不勾选时总延迟为 1 周期）。

配置完成，点击 OK 后选择想要的综合方式后，完成 IP 核创建。如果在配置

时没有修改 Component Name, 则 IP 核名称应为 blk_mem_gen_0。此时, 即可参照 blk_mem_gen_0_stub.v, 在项目其他部分中例化 IP 核以进行使用, 例化过程与 PLL 类似, 不再重复。关于该 IP 核模块接口的说明可见文末。

实验内容:

1. 构造 PLL

输入时钟使用 100MHz 系统时钟, 产生 200MHz 和 50MHz 两种时钟, 之后连接自己设计的 1:100,000,000 分频器后, (频率分别为 2Hz 和 0.5Hz) 连接到 FPGA 的 LED 灯管脚上, 上板观察。

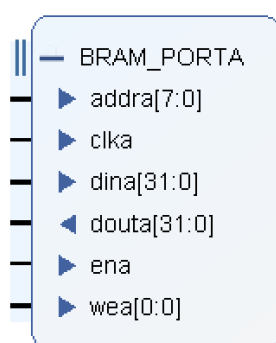
2. 构造 RAM (选做)

自行设计包含存储与读取的任务代码, 仿真观察其存储及读取功能。《数字逻辑与处理器基础》处理器大作业中提供了使用行为级描述书写的 RAM 模块 (DataMemory.v), 要求使用该手写 RAM 再次完成之前设计的任务 (注意两者接口与读取延迟上的不同), 比较并分析 Implementation 后的资源占用和时序性能。

附录:

1. Block Memory Generator 生成的单端口 RAM 接口说明

以示例中配置的 RAM 为例, 模块接口如图所示。

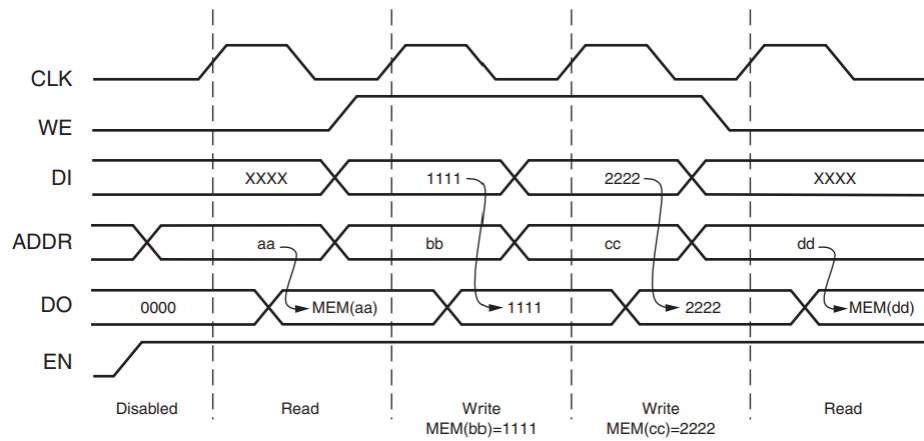


- ①DINA: 端口 A 写数据内容。
- ②ADDRA: 端口 A 读写地址。
- ③WEA: 端口 A 写使能信号, 高电平写, 低电平读。
- ④ENA: 端口 A 使能信号, 高电平使能, 低电平禁用端口。
- ⑤CLKA: 端口 A 的时钟信号。
- ⑥DOUTA: 端口 A 读数据内容。

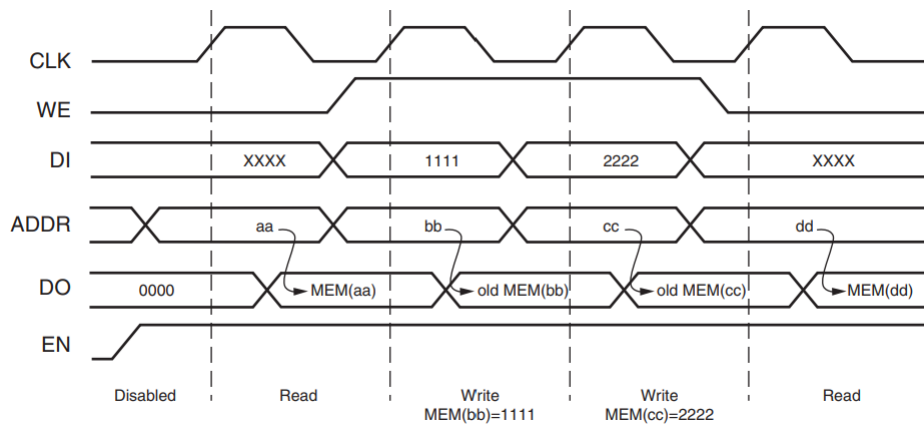
2. RAM 中的操作模式 (Operating Mode)

也称 Write Mode, 一共有三种, 决定了 RAM 读取数据的行为。

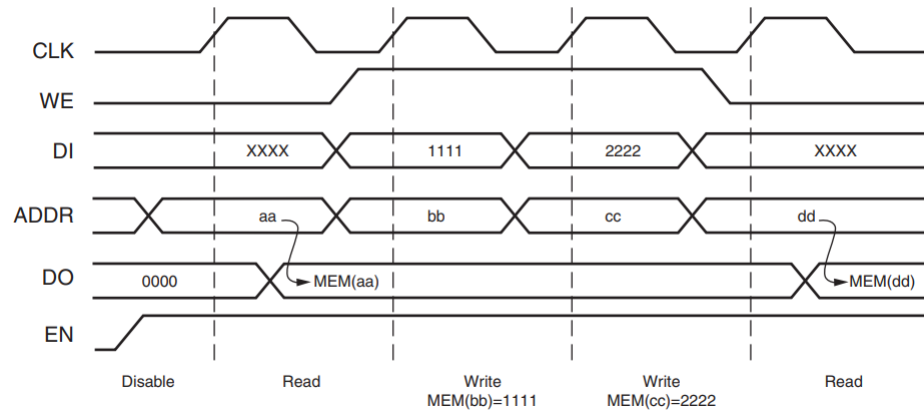
- ①Write First 写优先: 数据写入 RAM, 然后在下一个时钟输出该数据。



②Read First 读优先: 数据写入 RAM 中, 同时读出 RAM 中同地址的旧数据。



③No Change 不变: 读写彼此独立, 写入数据不影响读出结果。



有兴趣的同学可以参考 UG383, 进一步学习关于 Block RAM 的其他内容。