

# 架构向导和 IP 目录

## 介绍

现在的 Xilinx FPGA 包含了比基本的 LUT,CLB,I/OB 和布线更多的资源。与胶合逻辑发明时相比，现在 FPGA 被用来实现更复杂的数字电路。一些复杂的架构资源，例如时钟，必须利用配置和实例化，而不是自己实现。FPGA 工具也提供经常使用的复杂电路，例如 Reed-Solomon 译码器，使得开发者不需要“重新设计轮子”。本次实验介绍架构向导，和可以通过 IP 目录获得的 IP 生成工具。请参阅 Vivado 教程，了解如何使用 Vivado 工具创建项目和验证数字电路。

## 目标

完成此次试验后你将具备以下技能：

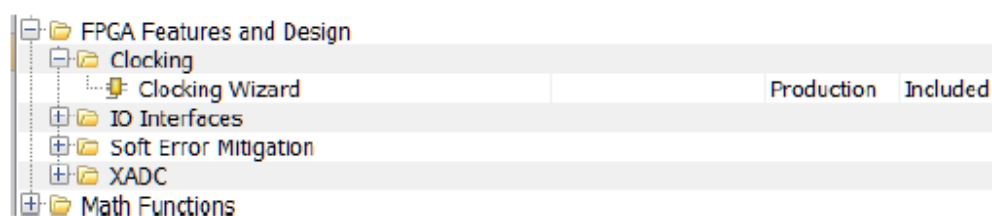
- 利用架构向导配置时钟资源
- 使用 IP 目录工具配置并使用计数器和存储器

## 架构向导

## Part1

通过正确的配置和实例化，而不是亲自实现，可以有效的使用专业和高级的架构。所使用的 FPGA 系列不同，所提供的架构的数量和种类也不同。在 Artix-7 系列中，架构向导提供时钟、SelectIO，软错误缓解和 XADC 资源。可以在 IP Catalog 工具的 FPGA Features and Design 文件夹下访问这些资源。

在 Basys3 和 Nexys4 DDR 板上，可以使用 100 MHz 时钟源，它连接到 FPGA 的 W5（Basys3）和 E3（Nexys4 DDR）引脚。该时钟源可用于产生多个不同频率和相移的时钟。这是通过使用称为 Artix-7 系列 FPGA 的数字时钟管理器（DCM）和锁相环（PLL）的架构资源来完成的。可以通过双击 IP Catalog 的 FPGA Feature and Design 文件夹的 Clocking 子文件夹下的 Clocking Wizard 条目来调用时钟源生成器。

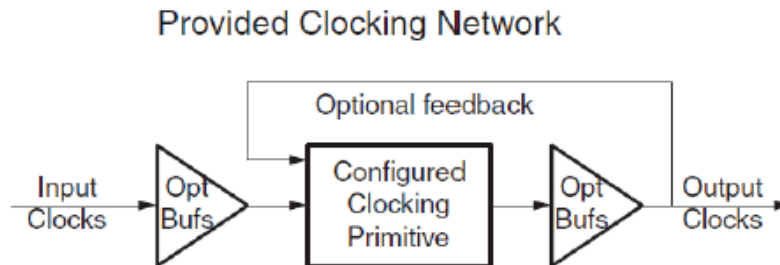


该向导可以轻松的你定制的时钟电路创建源代码封装器。该向导将指导你为时钟原语设置适当的属性，并允许你更改任何向导计算的参数。除了提供用于实现所需时钟电路的 HDL 包装器之外，时钟向导还提供由 Xilinx 时序工具为电路生成的时序参数汇总。该向导的主要功能包括：

- 每个时钟网络最多可接受两个输入时钟和七个输出时钟；

- 自动为所选设备选择正确的时钟原语；
- 根据用户选择的时钟功能自动配置时钟原语；
- 自动实现支持相移和占空比要求的整体配置；
- 可选择缓冲时钟信号。

时钟生成部件核心的功能如下图：



假设我们想要生成一个与 100 MHz 输入时钟同相的 5MHz 时钟。请按照以下步骤实现：

双击 **Clocking Wizard** 条目，向导打开后，你会看到有五个选项。

第一项标题为 Clocking Options。这里有与输入时钟，时钟参数，输入频率和范围相关的参数。由于实际的时钟频率就是 100MHz，所以我们保持默认值就行

第二项标题为 Output Clocks。可以设置输出时钟和所需频率相关的参数。设置输出频率为 1.000MHz。注意：如果频率显示为红色的表明存在错误。将鼠标移到该位置可以看到弹出提示“此设备的实际频率范围为 4.678MHz 至 800MHz”。现将其改为 5.000MHz。在输出时钟中取消勾选 **RESET** 选项，然后我们创建一个一步复位。

第三项标题为 MMCM Setting，用于显示计算设置。只要你知道他们都是做什么的或者它们是怎样影响设计的，你就可以检查并更改他们。我们希望看到时钟是稳定的。

第四项标题为 Port Renaming，允许你更改端口名。我们将使用默认的端口名。

最后的第五项标题为 Summary，显示你设置的摘要。

点击 OK 然后点击 Generate 生成用于综合，实现和仿真的文件。

可以通过“IP 源”选项卡访问文件(包括实例化文件)。以下是.vco 文件内容的示例。

```

clk_5MHz instance_name
    // Clock in ports
    .clk_in1(CLK_IN1), // IN
    // Clock out ports
    .clk_out1(CLK_OUT1), // OUT
    // Status and control signals
    .reset(RESET), // IN
    .locked(LOCKED)); // OUT
  
```

**1-1. 设计一个一秒脉冲发生器。使用时钟向导生成 5 MHz 时钟，通过时钟分频器（以行为建模编写）进一步分频，以生成一个以一秒为周期信号。上述使用时钟向导（以及生成的实例化模板）的步骤可用于本练习。使用 100 MHz 的板载时钟源，BTNU 按钮复位电路，SW0 作为使能信号，LED0 作为输出信号，Q 为生成的秒周期信号。LED15 用于输出 DCM 锁定信号。完成设计流程，生成比特流，并将其下载到 Basys3 或 Nexys4 DDR 板。验证功能。**

由于 Basys3 或 Nexys4 DDR 板上的 7 段数码管共用阴极，并且通过设置相应的阳极引脚来使数码管亮，因此需要扫描电路在多个数码管上显示信息（数字）。该电路应给比人眼可检测到的更快的频率循环驱动每一个数码管的阴极和相应的阳极。为了保证每个数码管持续且明亮的发光，所需的数字需要在 1 到 16ms 内驱动一次，刷新的频率为 1KHz 到 60Hz。如果刷新的速率慢到约 45Hz，大多数人将开始看到闪烁。

**1-2. 修改 Lab2\_2\_1（二进制到 BCD 转换器）的设计，将输入的 4 位二进制输入转化为的 BCD 值显示在两个 7 段显示器（而不是一个 7 段和一个 LED）上。使用 100 MHz 时钟源生成 5 MHz 时钟和相应的频率约为 500 Hz，用于驱动两个 7 段显示器的时钟分频器电路。生成比特流并将其下载到 Basys3 或 Nexys4 DDR 板以验证功能。**

## IP 目录

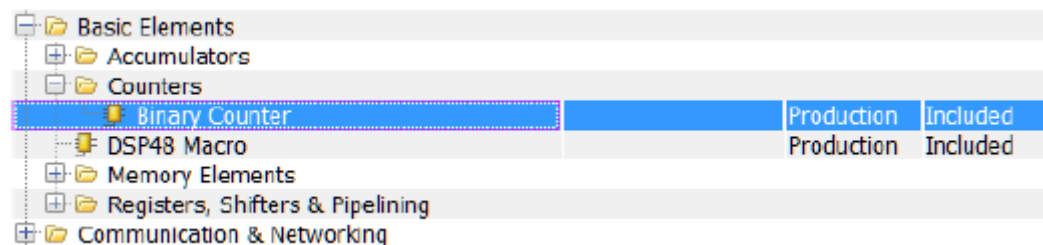
## Part 2

Vivado 工具的 IP 目录允许你配置和生成各种功能的核。在 IP 目录中，根据核心功能进行分组，这些核心包括从简单的基本核（如加法器）到非常复杂的和（如 MicroBlaze 处理

器)。它还涵盖了从汽车到图像处理等各个领域所需核。

配置和生成核心的过程与架构向导类似。核心将根据需要使用各种资源, 包括 LUT, CLB, DSP48, BRAM 等。让我们看看如何配置和生成计数器核。

二进制计数器核可以通过双击位于 IP catalog 下的 Basic Elements 分支下的 Counters 子文件夹下的 Binary Counter 来开始生成。



调用时, 您将看到两个项配置。第一项标题为 Basic, 其上的核心配置参数包括:

- Implement Using: Fabric or DSP48

- Output Width

- Increment Value

- Loadable, Restrict Count, Count Mode (Up, Down, UPDPWN), Threshold

第二项标题为 Control, 配置参数包括:

- Synchronous Clear, Clock Enable and various other settings.

设计人员可以选择想要的功能然后点击 **OK** 来生成 IP 核。

**2-1.使用 IP 目录生成一个简单的 4 位计数器内核, 从 0 计数到 9 (提示: 配置计数器核时使用 Threshold output) 将其实例化两次以创建已创建一个两位数的 BCD 计数器, 每秒计数一次。利用架构向导生成一个 5MHz 的时钟, 然后使用行为建模生成 1Hz 的周期信号来驱动计数器。将结果显示在两个 7 段数码管上。设计的输入使用 100MHz 的时钟源, 使用 BTNU 按钮作为复位信号, 使用 SW0 作为使能信号。使用 Basys3 或 Nexys4 DDR 板验证硬件设计功能。**

## 总结

本次实验中你学习了 Vivado 工具中的架构向导和 IP 目录。你已经使用架构向导生成了一个 5MHz 的时钟并用 IP 目录生成了一个计数器。IP 目录是一个功能强大的工具, 提供各种功能块, 可提高开发效率。