

第十五章 IP 核之 MMCM/PLL 实验

PLL 的英文全称是 Phase Locked Loop,即锁相环,是一种反馈控制电路。PLL 对时钟网络进行系统级的时钟管理和偏移控制,具有时钟倍频、分频、相位偏移和可编程占空比的功能。Xilinx 7 系列器件中的时钟资源包含了时钟管理单元 CMT,每个 CMT 由一个 MMCM 和一个 PLL 组成。对于一个简单的设计来说,FPGA 整个系统使用一个时钟或者通过编写代码的方式对时钟进行分频是可以完成的,但是对于稍微复杂一点的系统来说,系统中往往需要使用多个时钟和时钟相位的偏移,且通过编写代码输出的时钟无法实现时钟的倍频,因此学习 Xilinx MMCM/PLL IP 核的使用方法是我们学习 FPGA 的一个重要内容。本章我们将通过一个简单的例程来向大家介绍 MMCM/PLL IP 核的使用方法。

本章包括以下几个部分:

- 15.1 简介
- 15.2 实验任务
- 15.3 硬件设计
- 15.4 程序设计



15.1 简介

论坛:www.openedv.com

锁相环作为一种反馈控制电路,其特点是利用外部输入的参考信号控制环路内部震荡信号的频率和相位。因为锁相环可以实现输出信号频率对输入信号频率的自动跟踪,所以锁相环通常用于闭环跟踪电路。 锁相环在工作的过程中,当输出信号的频率与输入信号的频率相等时,输出电压与输入电压保持固定的相位差值,即输出电压与输入电压的相位被锁住,这就是锁相环名称的由来。

Xilinx 7 系列器件中具有时钟管理单元 CMT 时钟资源,xc7a35t 芯片内部有 5 个 CMT,xc7a100t 芯片内部有 6 个 CMT,为设备提供强大的系统时钟管理以及高速 I/O 通信的能力。时钟管理单元 CMT 的总体框图如下图所示。

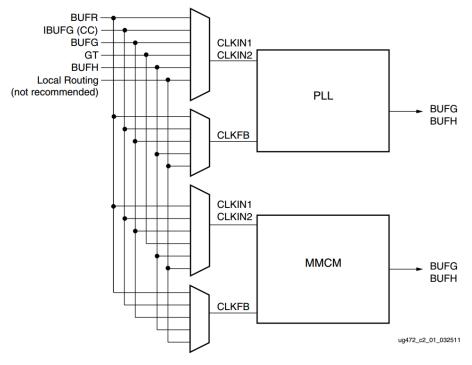


Figure 3-1: Block Diagram of the 7 Series FPGAs CMT

图 15.1.1 CMT 总体框图

MMCM/PLL 的参考时钟输入可以是 IBUFG(CC)即具有时钟能力的 IO 输入、区域时钟 BUFR、全局时钟 BUFG、GT 收发器输出时钟、行时钟 BUFH 以及本地布线(不推荐使用本地布线来驱动时钟资源)。在最多的情况下,MMCM/PLL 的参考时钟输入都是来自 IBUFG(CC)即具有时钟能力的 IO 输入,本实验也是如此。MMCM/PLL 的输出可以驱动全局时钟 BUFG 和行时钟 BUFH 等等。BUFG 能够驱动整个器件内部的通用逻辑的所有时序单元的时钟端口。BUFG/BUFH/CMT 在一个时钟区域内的连接框图如下图所示。



论坛:www.openedv.com

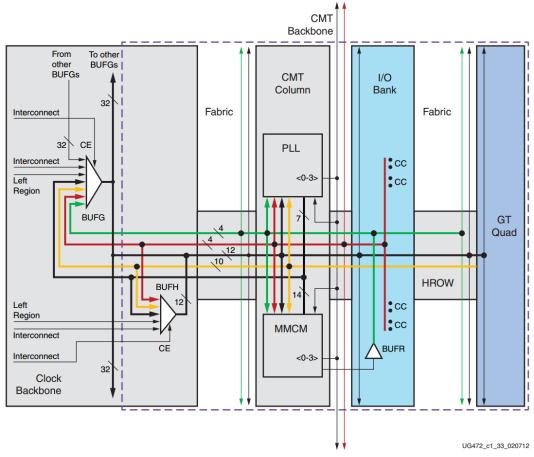


图 15.1.2 BUFG/BUFH/CMT 在一个时钟区域内的连接

在本实验中,读者可以简单地理解为:外部时钟连接到具有时钟能力的输入引脚 CCIO(Clock-Capable Input),进入 MMCM/PLL,产生不同频率和不同相位的时钟信号,然后驱动全局时钟资源 BUFG。但是要进行更深入的 FPGA 开发,就必须理解器件的时钟资源架构。有关 Xilinx 时钟资源和 CMT 的更详细信息,读者后期可以花一些时间和精力去学习 Xilinx 官方的手册文档"UG472,7 Series FPGAs Clocking Resources User Guide"里的介绍。

由上文介绍我们知道对于 Xilinx 7 系列的芯片一个 CMT 由一个 MMCM 和一个 PLL 组成。下面我们分别来看两者的关系。

②正点原子

原子哥在线教学: www.yuanzige.com

论坛:www.openedv.com

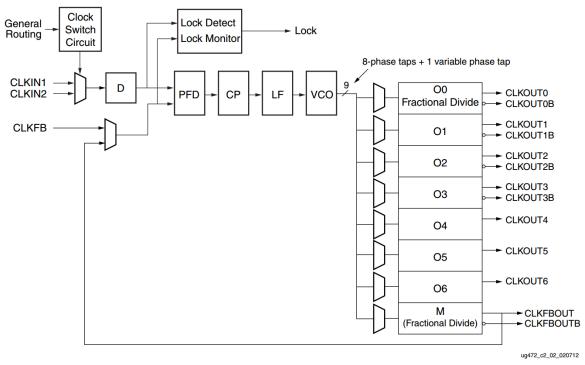


Figure 3-2: Detailed MMCM Block Diagram

图 15.1.3 MMCM 总体框图

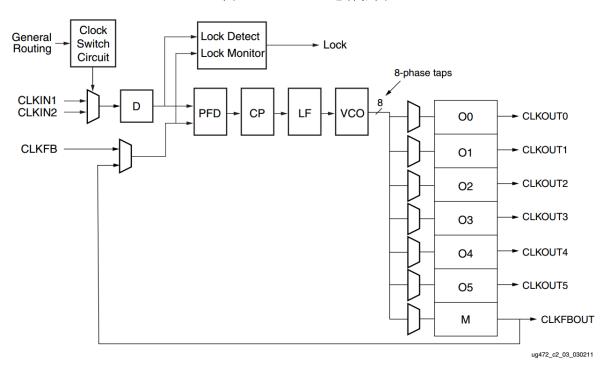


Figure 3-3: Detailed PLL Block Diagram

图 15.1.4 PLL 总体框图

由两张图我们可以直观看到MMCM和PLL的组成结构基本相同。

MMCM (Mixed-Mode Clock Manager) 混合模式时钟管理器,MMCM 功能是 PLL 的超集,它是在 PLL 的基础上加了相位动态调整功能,PLL 是模拟的,而动态调相是数字电路,所以称为混合模式。其相对于



论坛:www.openedv.com

PLL 的优势是相位可以动态调整,占用面积较大。MMCM 主要用于驱动器件逻辑(CLB、DSP、RAM 等)的时钟。PLL 是和 IO 资源紧密绑定的,占用面积小,常用于为内存接口生成所需的时钟信号,但也具有与其他器件逻辑的连接,因此如果需要额外的功能,它们可以用作额外的时钟资源。关于 PLL 的工作原理我们简单概括如下:

PLL 由以下几部分组成: 前置分频计数器(D 计数器)、相位频率检测器(PFD, Phase-Frequency Detector) 电路、电荷泵(Charge Pump)、环路滤波器(Loop Filter)、压控振荡器(VCO, Voltage Controlled Oscillator)、反馈乘法器计数器(M 计数器)和后置分频计数器(O1-O6 计数器)。

在工作时,PFD 检测其参考频率(F_{REF})和反馈信号(Feedback)之间的相位差和频率差,控制电荷泵和环路滤波器将相位差转换为控制电压; VCO 根据不同的控制电压产生不同的震荡频率,从而影响 Feedback 信号的相位和频率。在 F_{REF} 和 Feedback 信号具有相同的相位和频率之后,就认为 PLL 处于锁相的状态。

在反馈路径中插入M计数器会使 VCO 的震荡频率是 F_{REF} 信号频率的 M 倍, F_{REF} 信号等于输入时钟 (F_{IN}) 除以预缩放计数器 (D) 。参考频率用以下方程描述: F_{REF} = F_{IN} /D, VCO 输出频率为 F_{VCO} = F_{IN} *M/D, PLL 的输出频率为 F_{OUT} = $(F_{IN}$ *M) / (N*O) 。

Xilinx 提供了用于实现时钟功能的 IP 核 Clocking Wizard,该 IP 核能够根据用户的时钟需求自动配置器件内部的 CMT,以实现用户的时钟需求。在这里我们主要讲解的是如何使用该 IP 核,有关该 IP 核的更详细介绍,读者可以参阅 Xilinx 官方的手册文档"PG065,Clocking Wizard v5.2 LogiCORE IP Product Guide"。

15.2 实验任务

本节实验任务是使用开发板输出 1 个时钟频率,并在 Vivado 中进行仿真以验证结果,最后生成比特流文件并将下载到开发板上。

15.3 硬件设计

本章实验将 Clocking Wizard IP 核产生 1 个时钟 10MHz。



15.4 程序设计

我们首先创建一个工程,接下来添加 PLL IP 核。在 Vivado 软件的左侧 "Flow Navigator"栏中单击"IP Catalog"按钮以及单击后弹出的"IP Catalog"窗口如下图所示。

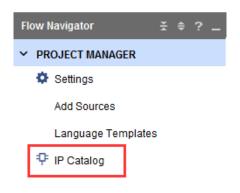


图 15.4.1 "IP Catalog" 按钮

打开 "IP Catalog"窗口后,在搜索栏中输入"clock"关键字,可以看到 Vivado 已经自动查找出了与关键字匹配的 IP 核名称,如下图所示。

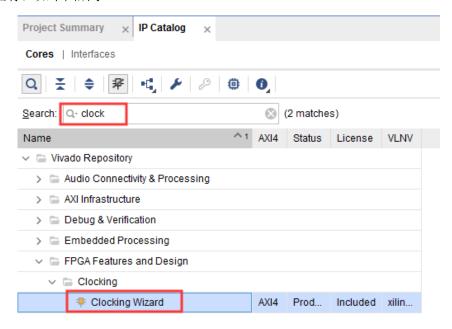


图 15.4.2 搜索栏中输入关键字

我们双击 "FPGA Features and Design" → "Clocking" 下的"Clocking Wizard",弹出"Customize IP" 窗口,如下图所示。





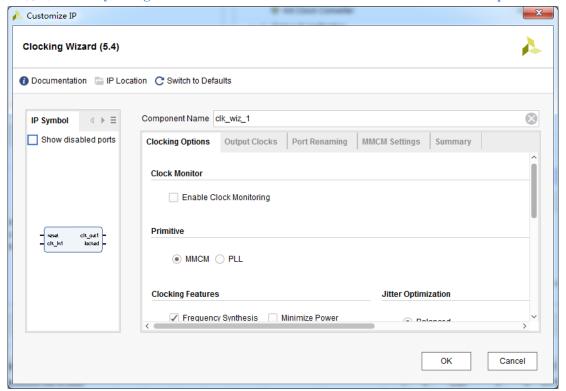


图 15.4.3 "Customize IP" 窗口

接下来就是配置 IP 核的时钟参数。最上面的"Component Name"一栏设置该 IP 元件的名称,这里保持默认即可。在第一个"Clocking Options"选项卡中,"Primitive"选项用于选择是使用 MMCM 还是 PLL 来输出不同的时钟,对于我们的本次实验来说,MMCM 和 PLL 都可以完成,这里我们可以保持默认选择 MMCM。需要修改的是最下面的"Input Clock Information"一栏,把"Primary"时钟的输入频率修改为我们开发板的开发板上的晶振频率 100MHz,其他的设置保持默认即可,如下图所示。





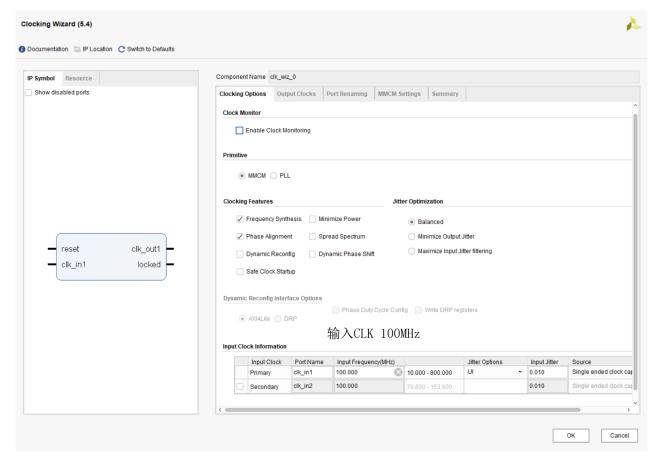
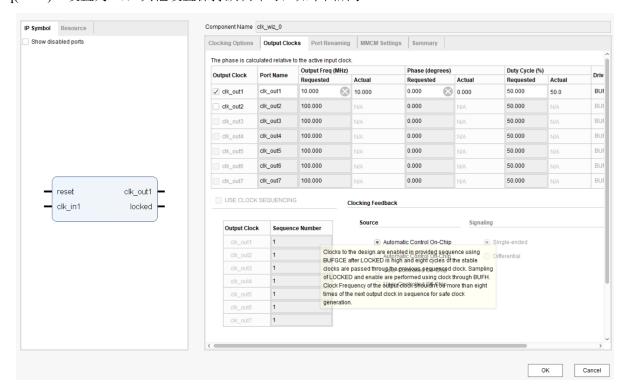


图 15.4.4 "Clocking Options"选项卡的设置

接下来切换至"Output Clocks"选项卡,在"Output Clock"选项卡中,勾选第1个时钟,并且将其"Output Freq(MHz)"设置为10,其他设置保持默认即可,如下图所示。





论坛:www.openedv.com

"Port Renaming"选项卡主要是对一些控制信号的重命名。这里我们只用到了锁定指示 locked 信号,其名称保持默认即可,如下图所示。

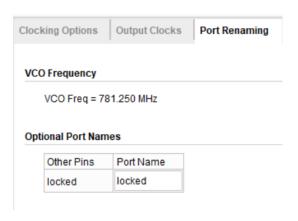


图 15.4.6 "Port Renaming"选项卡的设置

"MMCM Setting"选项卡展示了对整个 MMCM/PLL 的最终配置参数,这些参数都是根据之前用户输入的时钟需求由 Vivado 来自动配置,Vivado 已经对参数进行了最优的配置,在绝大多数情况下都不需要用户对它们进行更改,也不建议更改,所以这一步保持默认即可,如下图所示。

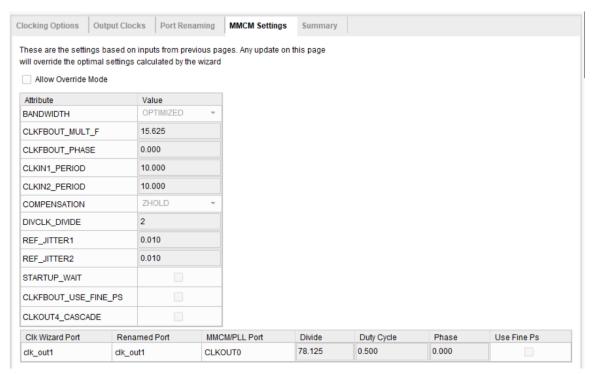


图 15.4.7 "MMCM Setting"选项卡的设置

最后的"Summary"选项卡是对前面所有配置的一个总结,在这里我们直接点击"OK"按钮即可,如下图所示。



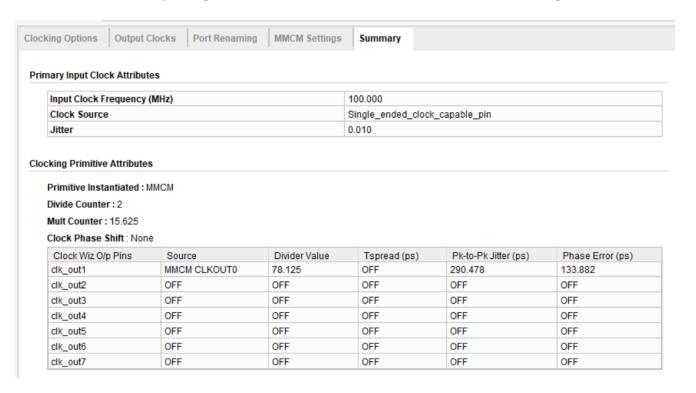


图 15.4.8 "Summary" 选项卡

接着就弹出了"Generate Output Products"窗口,我们直接点击"Generate"即可,如下图所示。

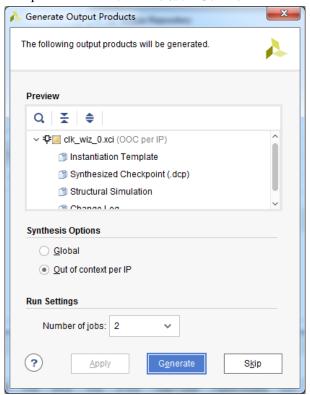


图 15.4.9 "Generate Output Products"窗口

之后我们就可以在"Design Run"窗口的"Out-of-Context Module Runs"一栏中出现了该 IP 核对应的run"clk wiz 0 synth 1", 其综合过程独立于顶层设计的综合, 所以在我们可以看到其正在综合, 如下图



论坛:www.openedv.com

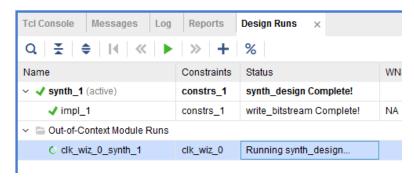


图 15.4.10 "clk wiz 0 synth 1"run

在其Out-of-Context综合的过程中,我们就可以开始编写代码了。首先打开IP核的例化模板,在"Source"窗口中的"IP Sources"选项卡中,依次用鼠标单击展开"IP"-"clk_wiz_0"-"Instantitation Template",我们可以看到"clk_wiz.vho"文件,它是由 IP核自动生成的只读的VHDL例化模板文件,双击就可以打开它,在例化时钟 IP核模块的时钟,可以直接从这里拷贝,如下图所示。

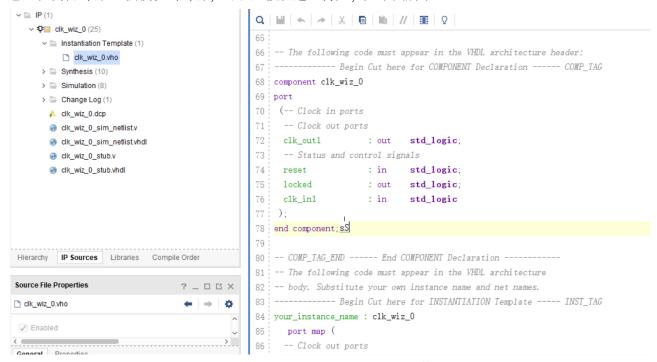


图 15.4.11 "clk wiz.vho" 文件