

PLL 配置详细说明

PLL 的配置需求

假定设计者已经新建了一个工程，然后需要配置一个 PLL。该 PLL 的输入时钟为 FPGA 外部的 25MHz 晶振，希望得到一个 50MHz（输入时钟的 2 倍频）的系统时钟供 FPGA 内部使用。该 PLL 的输入输出接口如表 1 所示。

表 1 PLL 的接口定义

信号名	方向	功能描述
inclk0	input	PLL 输入时钟
areset	input	PLL 复位信号，高电平有效
c0	output	PLL 输出时钟
locked	output	该信号用于指示 PLL 处理后的时钟已经稳定输出，高有效

PLL 的配置步骤

① 如图 1 所示，在 Quartus II 的菜单栏选择“Tools—>MegaWizard Plug-In Manager…”。

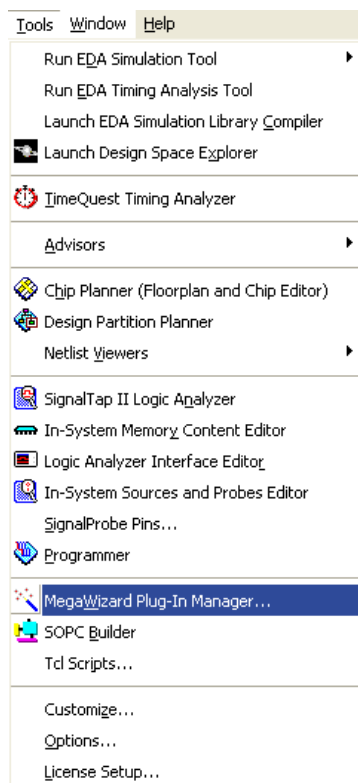


图 1 选择 MegaWizard

② 如图 2 所示，使用默认选项“Create a new custom megafunction variation”，

点击 “Next>”。

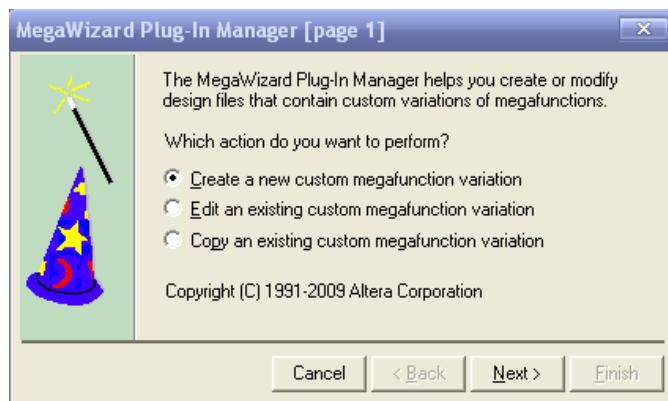


图 2 新建 megagunction

③ 如图 3 所示，进行以下配置：

- 在 “Select a megafunction from the list below” 窗口内打开 “I/O” 下拉框，选择 “ALTPLL”。
- 在 “Which type of output file do you want to create?” 下选择 “Verilog HDL”，这是配置的 PLL 内核使用的语言，一般选择此项。
- 在 “What name do you want for the output file?” 里默认会出现当前设计的工程路径，需要设计者在最后面手动输入例化的 PLL 的名字，这里输入了 “PLL_ctrl”。

完成以上配置，点击 “Next>”。

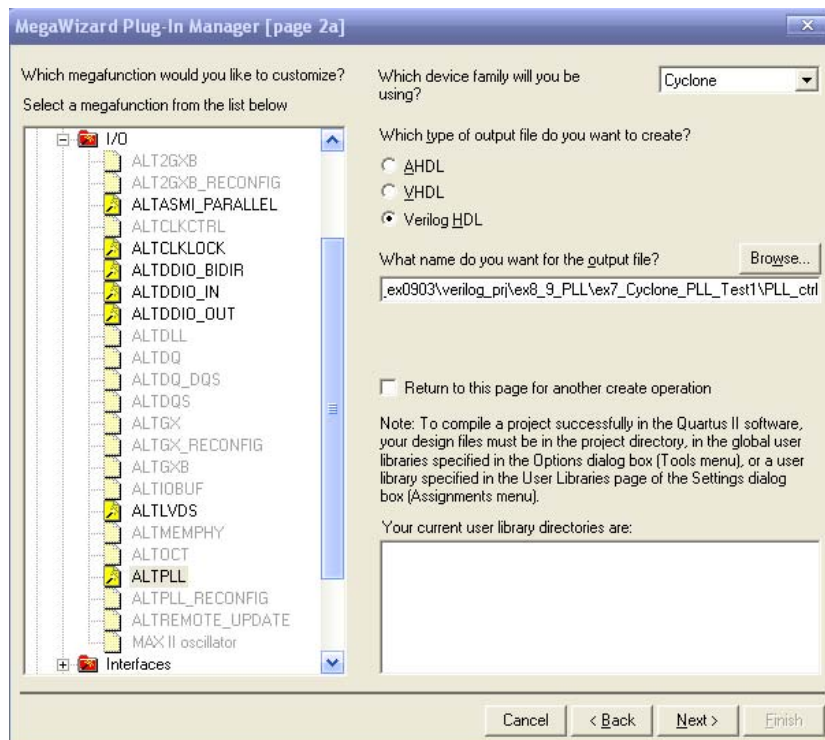


图 3 新建 PLL

④ 如图 4 所示，进行以下配置：

- 在“General”一栏内的“Which device speed grade will you be using?”选则该工程所使用器件的速度等级。
- 在“What is frequency of the inclock0 input?”内选择 PLL 输入时钟的频率。

其他选项使用默认即可。点击“Next>”。

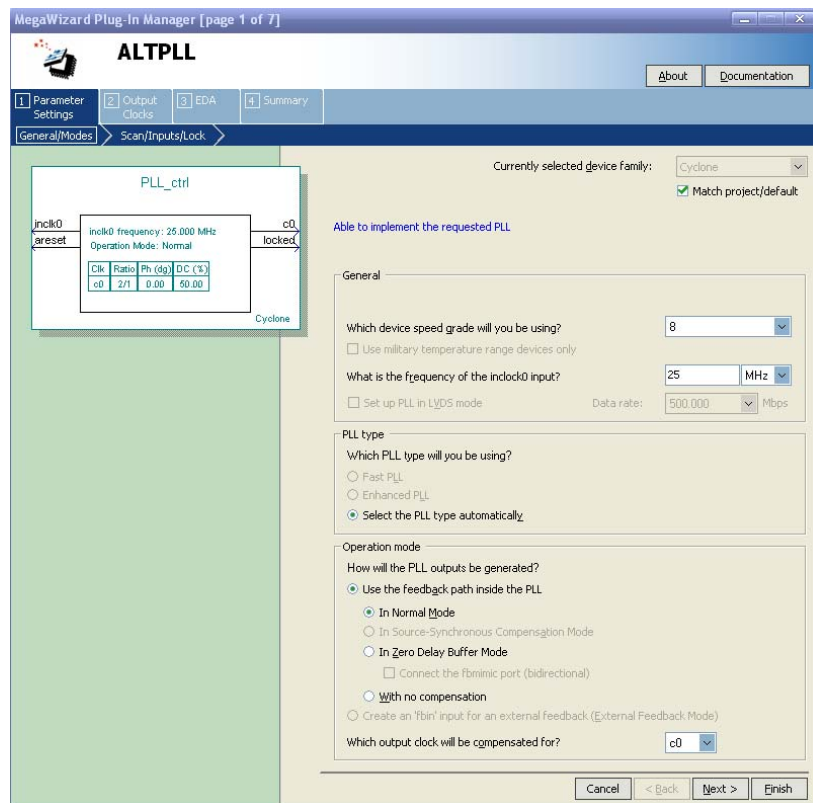


图 4 输入时钟配置

⑤ 如图 5 所示，配置如下：

- 在“Option input”一栏内勾选“Creat an ‘areset’ input to asynchronously reset the PLL”。
- 在“Lock output”中勾选“Creat ‘locked’ output”。

其他选项使用默认即可。点击“Next>”。

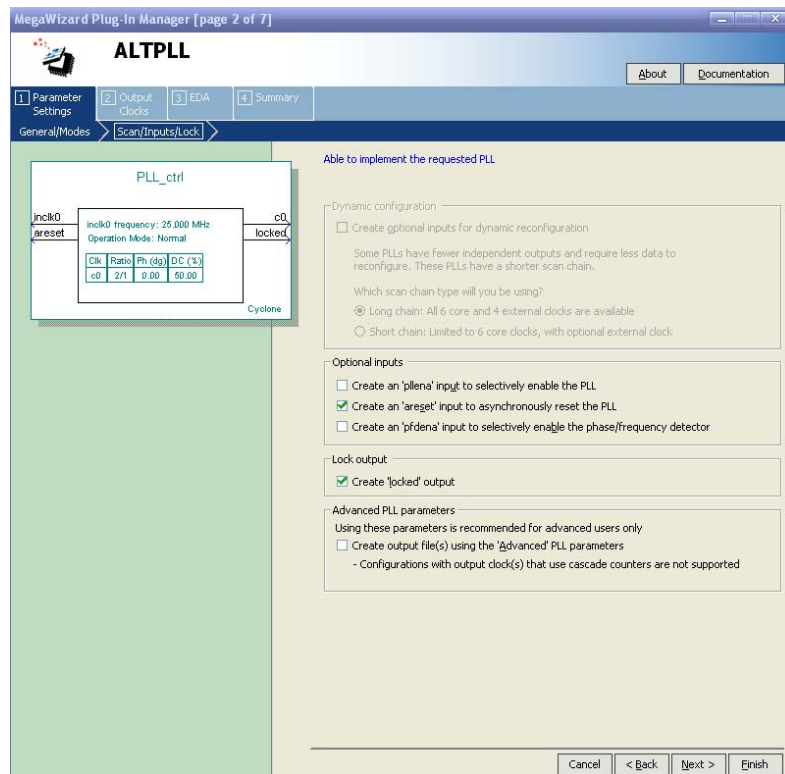


图 5 配置控制信号

⑥ 配置输出时钟 c0 相关参数，如图 6 所示。

- 设计者可以在“Enter output clock frequency?”后面输入希望得到的 PLL 输出时钟的频率。

设计者也可以在“Enter output clock parameter?”后面设置相应的输出时钟和输入时钟的频率关系。“Clock Multiplication factor”后输入倍频系数，“Clock division factor”后输入分频系数，二者决定了输出时钟频率。

- 在“Clock phase shift”中可以设置相位偏移。
- 在“Clock duty cycle”中可以设置输出时钟占空比。

按照图 6 设置后，点击“Next>”。

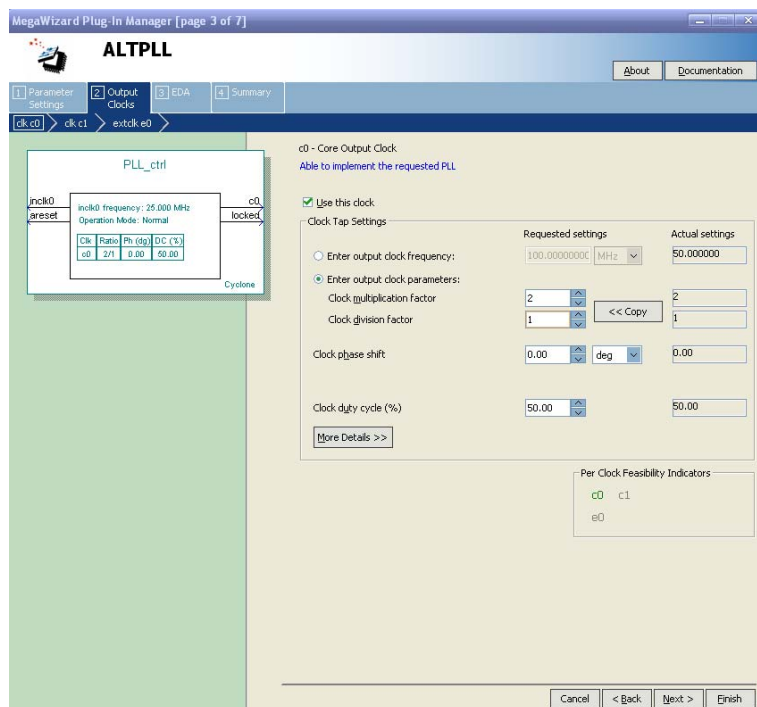


图 6 配置输出时钟 c0

⑦ “clk c1” 选项是可选的，用户需要第二个输出时钟时可以开启该输出时钟，相应勾选 “Use the clock” 后和上一步类似进行配置即可。点击 “Next>”。

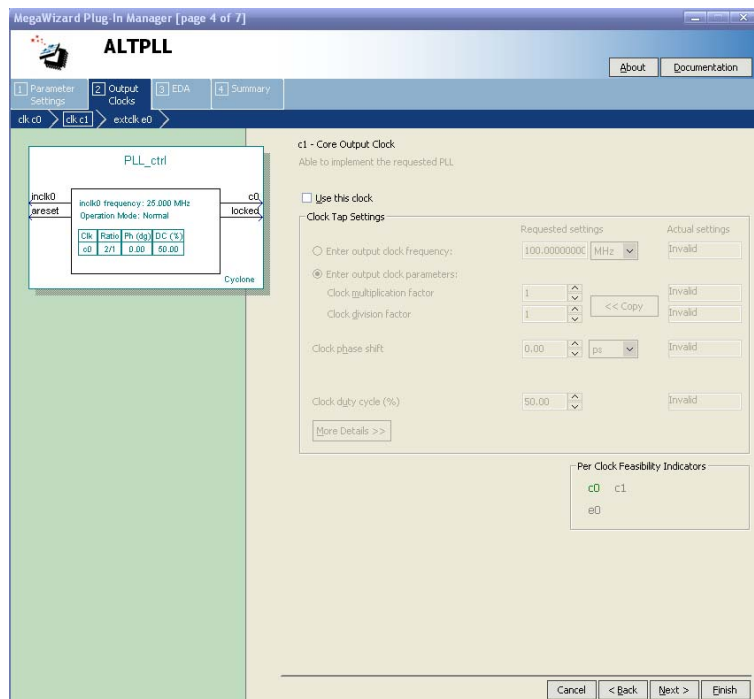


图 7 配置输出时钟 c1

⑧ “extclk e0” 也是可选的，该时钟主要是输出给 FPGA 外部器件作为时钟，不能作为内部时钟使用。用户需要该输出时钟时可以开启该输出时钟，相应勾选 “Use the clock” 后和前一步类似进行配置即可。点击 “Next>”。

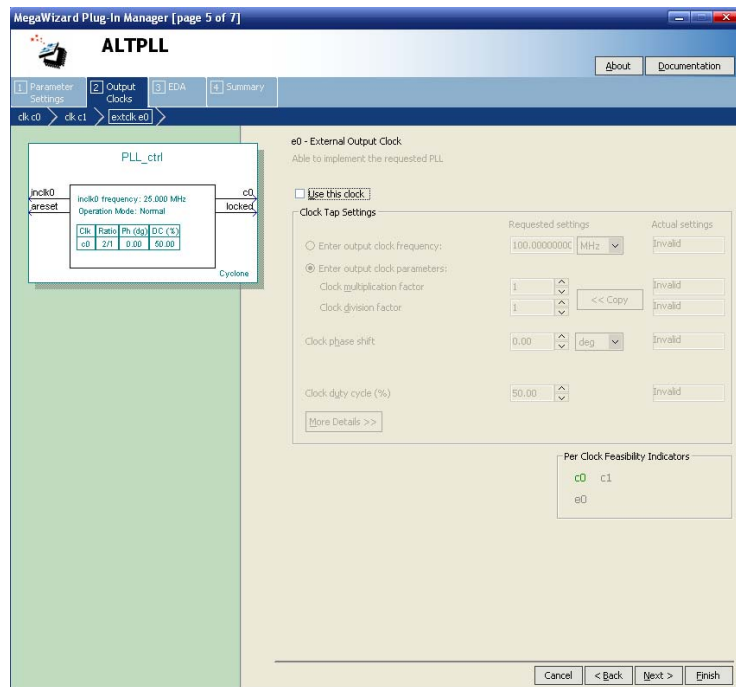


图 8 配置输出时钟 e0

⑨ 如图 9 所示，“EDA”中列了用户在对例化了 PLL 模块的工程仿真时，需要添加的仿真库文件，用户可以到 Quartus II 安装文件夹下可以找到。点击“Next>”。

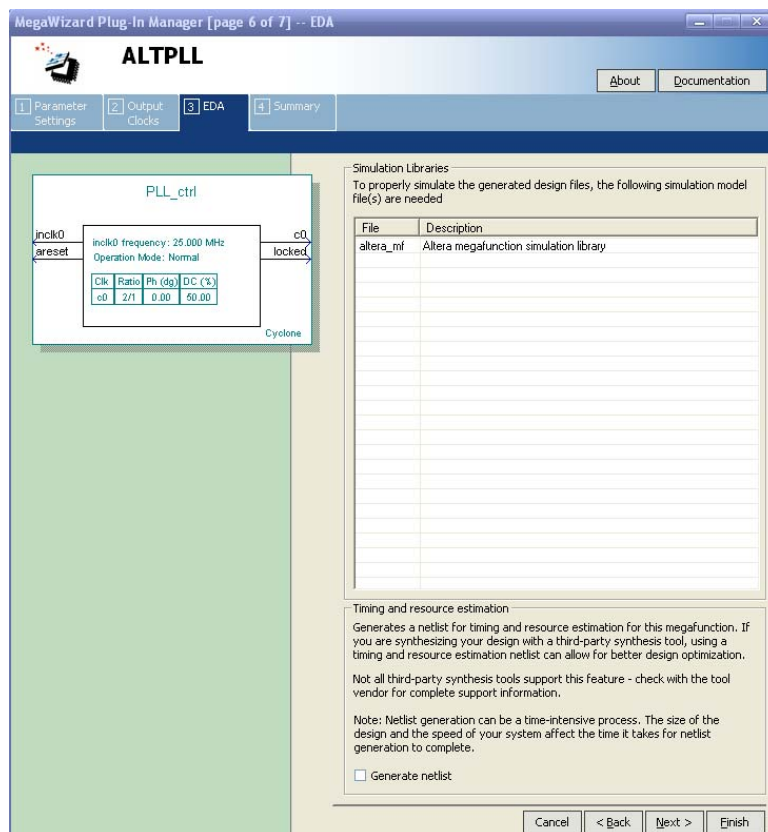


图 9 仿真库文件

⑩ 如图 10 所示，“Summary”中罗列了该 PLL 核最终的输出文件。对主要的一些输出文件说明如下：

- PLL_ctrl.v, 字面翻译是“变异文件”, 是 PLL 内部的控制 IP 核。
- PLL_ctrl_inst.v 是一个模板的例化文件, 用户可以直接复制这个文件里的例化来用。
- PLL_ctrl_wave.jpg 里是用户所配置的 PLL 的波形示例, 勾选后可以在工程目录下找到, 大家可以就我们的工程去看看波形是否符合预定的要求。或者用它和仿真后的波形对比一下, 它们应该是一致的。

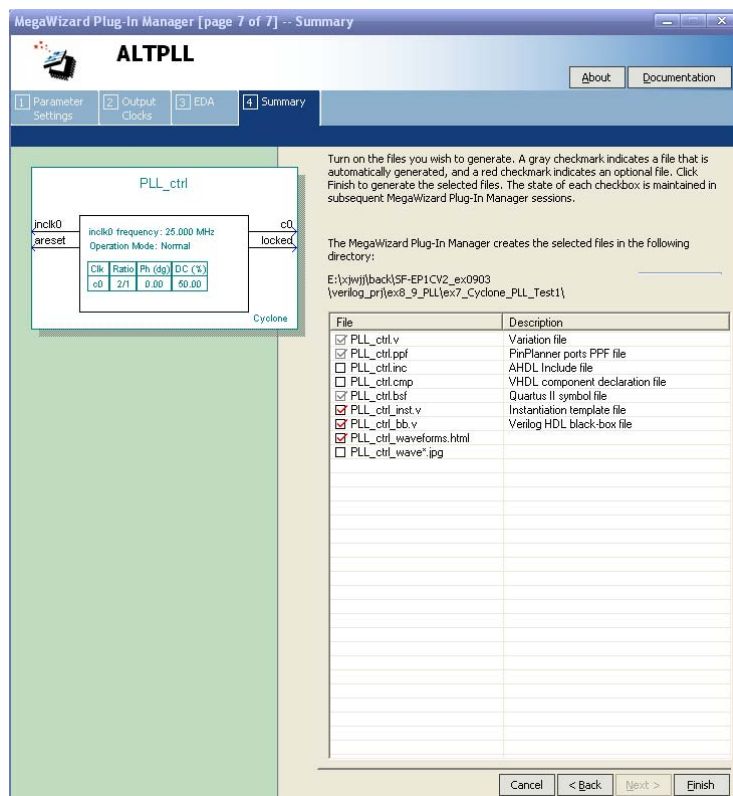


图 10 输出文件

PLL 的例化

PLL 配置完成后, 需要将 PLL 例化到工程中。找到“PLL_ctrl_inst.v”文件并打开, 将其内容拷贝到工程代码中, 然后更改“()”里的内容, 使用代码外的接口对应即可。如:

```
input clk;          //25MHz 系统外部输入时钟
input rst_n;        //系统复位信号, 低电平有效

output clkdiv;      //PLL 输出时钟
output locked;      //稳定 PLL 输出标志位, 高有效

//PLL 产生模块
//产生一个系统输入时钟 2 倍频, 相移 0 度的时钟
PLL_ctrl PLL_ctrl_inst (
```

```
.areset(~rst_n),    //PLL 异步复位信号, 高有效
.inclk0(clk),       //PLL 输入时钟
.c0(clkdiv),        //PLL 输出时钟
.locked(locked)     //稳定 PLL 输出标志位, 高有效
);
```

最后，设计者完善代码的其他部分，编译工程即可。