

# 提升创造力的数字设计工具： FPGA Editor

赛灵思资深现场应用工程师 Clayton Cameron 与您分享 ISE 工具套件中他最喜欢的工具 FPGA Editor 的使用技巧



作者：  
Clayton Cameron  
赛灵思公司 (Xilinx, Inc.) 资深现场应用工程师  
[clayton.cameron@xilinx.com](mailto:clayton.cameron@xilinx.com)

工程师在设计过程中，经常需要一定的创造力(你不妨称之为数字管道胶带)才能够保证设计的顺利完成。过去8年时间里，我曾经目睹许多优秀工程师利用这一方法出色地完成了许多工作，而他们采用的最主要工具就是 FPGA Editor。

利用 FPGA Editor，你可以察看完成的设计并确定是否在FPGA构造一级真正实现了设计意图 – 而这对于任何工程师或现场应用工程师来说都是非常需要的。假设你拿到协作者的设计，需要对其进行修改，但他们的 HDL 源代码非常难于理解，或者根本没有任何注释或文档。也许你只是想将某些时钟逻辑锁定，但却不知道实例的名称或者如何将其锁在所希望的位置。下面讨论的一些用来探查 FPGA 构造以及创建命令行补丁的技巧可以帮助设计人员满足不断逼近的任务期限要求。

## FPGA 构造勘察技巧

当赛灵思发布针对新 FPGA 器件的工具时，通常我做的第一件事就是

打开 FPGA Editor 来勘 FPGA 器件的内部构造。具体做法是找到 **Xilinx** → **ISE** → **Accessories** 菜单并点击 **FPGA Editor** 图标, 或者在命令行键入 `fpga_editor`。图形用户界面打开后, 在 **File** 菜单下选择 **New**。FPGA Editor 会询问设计文件名称和物理约束文件。此时还没有任何设计文件, 因此随便键入一个设计文件名(例如, `test.ncd`), 并选择希望察看的器件类型。FPGA Editor 将会使用同样的名字作为物理约束文件的名称, 并加载一个空白设计。

另一种方法是编译随软件提供的某个 ISE® 工具套件设计例子, 并将其加载到 FPGA Editor 中来察看 FPGA 构造。加载一个设计实例可以显示更多细节, 并且更容易定位感兴趣的项目。

在 FPGA Editor 中浏览只需要记住两件事:

- 1、如何利用 **CTRL / Shift** 快捷键进行放大缩小。
- 2、如果利用 **F11** 键放大选定的项目。

不使用 GUI 按钮快速缩放的方法是, 按住 **Ctrl** 和 **Shift** 键, 然后分别利用鼠标左键和右键进入放大和缩小。要想快速找到任何项目, 在 GUI 右上角的列表窗口(List window) 中选择该项目即可。一旦定位了想找的项目, 按 **F11** 键。阵列窗口(Array window) 就会放大显示该项目。

FPGA Editor 有四个主要窗口: 列表(List)、全局(World)、阵列(Array) 和块(Block)。List窗口显示设计中所有活动的项目。通过此窗口顶部的下拉菜单可选择其内容 – 列表内容包括已经布局或还未使用的部件、网络或未布线的网络等等。

全局视图窗口(World Windows) 始终显示完整 FPGA 硅片视图, 这在试图确定某个网络的布线情况时非常有

## 利用FPGA Editor, 你可以察看物理实施后的设计并确定是否在FPGA构造一级真正实现了设计意图 – 而这对于任何工程师或现场应用工程师来说都是非常需要的。

用。同时, 阵列窗口(Array window) 则是 FPGA 构造和逻辑的动态视图。如果双击 Array 视图中的任何项目, 会显示 Block 视图, 给出所选择项目或逻辑单元的详细情况。

为方便浏览或对设计进行编辑, 这些窗口可以复制。许多情况下, 打开第二个 Array 窗口会比较方便同时在设计的两个不同部分工作。例如, 假设需要在全局时钟缓冲器和芯片底部的触发器之间增加一条布线。如果在一个 Array 窗口中显示全局时钟缓冲器的输出, 另一个 Array 窗口则显示触发器时钟输入, 那么会做起来会更方便。不然的话就必须来回缩放来定位布线的两端, 显然这很繁琐。

在 FPGA Editor GUI 的右侧是一组由 20 个功能按钮组成的工具条, 用于设计察看和编辑。通过编辑 `$XILINX/data` 目录下的 `fpga_editor.ini` 文件可以增加更多功能按钮。当察看设计时, 应当随时使用 **INFO** 按钮。该功能会将所选择项目的信息转录到控制台(Console) 窗口。这一功能非常方便, 你可以在控制台窗口中标注数据并将其拷贝到其它地方使用, 例如编写 UCF 约束。

一旦了解了最基本的概念和操作, 就可以开始察看 FPGA 构造。通常我都从时钟逻辑开始。这包括数字时钟管理器(DCM)、锁相环(PLL)、全局时钟缓冲器(BUFG)、区域时钟缓冲器(BUFR)、I/O缓冲器(BUFIO) 以及不同的时钟区域。(要想按字母列出项目表,

可以到 LIST 窗口点击 Type 来排序。) 点击一个 **DCM** 并按 **F11**。ARRAY 窗口会定位到选择的 DCM 并放大显示它。继续点击 **DCM** 并观察 GUI 底部的 Console 窗口, 其中会显示与下面类似的内容:

```
comp "DCM_BASE_inst_star", site
"DCM_ADV_X0Y9", type = DCM_ADV
(RPM grid X73Y202)
```

这是有用的数据。拷贝并粘贴上述内容到 UCF 文件中, 并作如下更改以锁定这一 DCM 逻辑:

```
INST "DCM_BASE_inst_star"
LOC=DCM_ADV_X0Y9;
```

利用这一方法, 几乎可以锁定 FPGA 中的任何项目。下面是 BUFG 锁定的另一个例子:

```
comp "BUFG_inst_star", site
"BUFGCTRL_X0Y20", type = BUFG
(RPM grid X73Y124)
```

```
INST "BUFG_inst_star"
LOC=BUFGCTRL_X0Y20;
```

再次回到 List 窗口并标注同一 DCM。双击之后将会在 Block 视图中显示该 DCM 以及所有设置和参数。这是一项非常强大的功能, 可用于 FPGA 构造中的任何逻辑项目。如果选择一个逻辑片并双击它, 就可以看到逻辑片是如何布线连接的, 以及是否使用了进位链或本地触发器。

Block 视图的按钮条包含许多其它选项。值得一提的是 **F= button** 按钮,

其功能是显示逻辑片中使用的项目的完整配置。例如，如果使用了一个 LUT6 和一个触发器，按下 **F=** 按钮将会给出 LUT 的布尔议程以及触发器的配置模式。

阅读赛灵思用户指南是一回事，而从计算机屏幕上展开的视图上探索所有逻辑、开关和参数则是另一回事。一旦熟悉了 FPGA 构造及所有资源，那么在编写和验证设计时获得的帮助将会令人感到惊奇。

### 在设计流程中记录生成补丁脚本

当用户在 GUI 环境中编辑设计时，FPGA Editor 能够记录用户动作。用户不仅可以保存记录动作流程，还可在以后重新使用记录的脚本并加以重复。当无法更改 RTL，但又需要在设计过程中对设计进行修改时，这一功能特别有用。假设设计采用了第三方 IP 或赛灵思加密 IP，其中包括一个全局时钟和一个 DCM 来生成称为 `interface_clk` 的时钟。然后再假设接口所连接的 ASIC 出现问题，无法在预定的 `interface_clk` 的上升沿接收数据。如何修正这一问题？

当然，你可以改变 PCB，更换出现问题的 ASIC 或者让第三方 IP 小组来修改时钟输出逻辑以提供 90 度相移的 `interface_clk`。所有这些解决方案都既耗费时间又成本高昂。一个更简单的建议是利用 FPGA Editor 来记录修改动作，对 `interface_clk` 逻辑进行必要的更改，从而为出现问题的 ASIC 提供正确的时钟相位。一旦有了相关修改的 FPGA Editor 脚本，就可以回放这些命令行脚本记录的修改步骤，而你也可以正常继续你的 FPGA 设计流程。当出现问题的 ASIC 修复正常以后，你只需要将 FPGA Editor 脚本从编译脚本中移除就可以，而 `interface_clk` 也会恢复其正常的行为。

要想对设计进行手动编辑，首先需要要在 FPGA Editor 中开启读/写 (read/

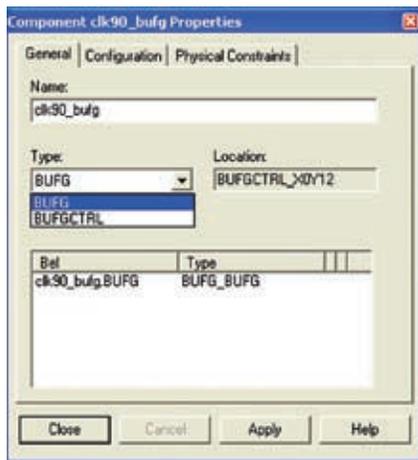


图 1 - 用户可以在属性窗口中配置和命名所选逻辑项。

write) 权限。在菜单条中点击 **File** → **Main Properties** 在此菜单下，可以调整编辑模式 (从 **No logic Change** 到 **Read/Write**)。点击 **Apply**，现在就可以开始对设计进行编辑了。在下一步利用 FPGA Editor 记录对设计进行的所有修改时，只需要简单地从菜单条中点击 **Tools** → **Scripts** → **Begin Recording**。FPGA Editor 将会提示输入一个脚本名字 (如 `patch.scr`)。输完脚本名字，就可以对设计进行必要的修改了。

在设计中运行设计规则检查(DRC)

来看一下是否有规则冲突的红色标志是一个很好的方式。在我的设计例子中，有 14 条警告，但都可以忽略。下一步我们将需要定位 `interface_clk` 使用的 DCM，并为此 DCM 的 90 度相移输出创建另一个称为 `DCM_clk90_out` 的时钟。这需要利用全局时钟布线资源将时钟连接到 BUFG。要增加一个 BUFG，先在 FPGA 构造中找到一个未用的 BUFG 位置，右击并选择 **Add** (添加)。然后，工具会提醒你为 BUFG 命名 (`clk90_bufg`) 并确定其类型: BUFG (参见图 1)。

创建了新的 BUFG，就需要将其输入和输出连接到适当的位置。在本例中，DCM 的 90 度相移输出将驱动 BUFG。在窗口 Array1 中，点击 DCM 的 90 度输出端衰减器，在窗口 Array2 中，点击 BUFG 的输入端衰减器，同时按住 **Ctrl** 键，可实现连接。然后释放 **Ctrl** 键，点击鼠标右键并选择 **Add**。工具会提示为新网络连接输入名字。这样就将 DCM 和 BUFG 通过新网络连接到一起了 (参看图 2)。

`clk90_bufg` 的输出要代替由原

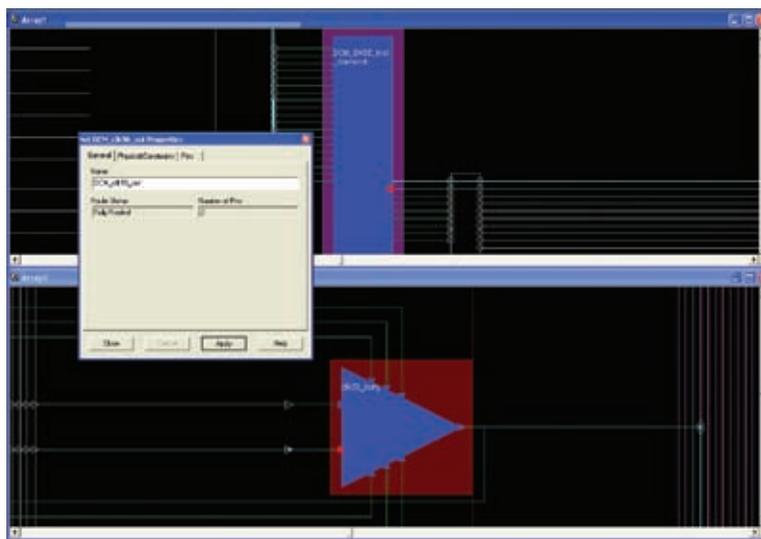


图 2 - 在两个逻辑项之间手动布线时，使用两个“阵列”窗口便于选择线路首尾，如红三角所示。

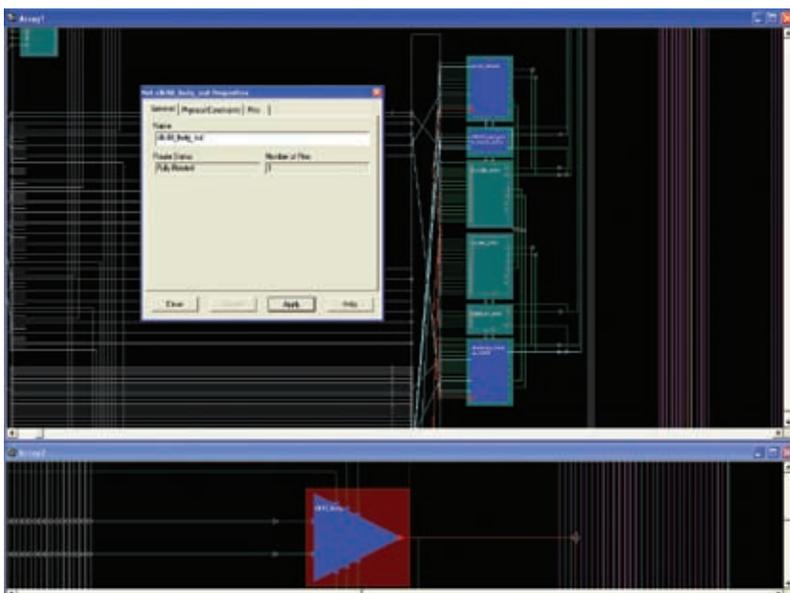


图3-BUFG 输出网络属性窗口显示网络连接数和完全布线的网络状态。

interface\_clk 驱动的 IOB 上的时钟。要将 IOB 从原来的时钟域中移开，需要定位 IOB，标注时钟输入端衰减器并按下 **Delete** 键移除这一连接。这样我们就可以将新的 clk90\_bufg 时钟连接到这儿，从而完成整个补丁。要连接 BUFG (clk90\_bufg) 的输出，按住 **Ctrl** 键不放，在窗口 Array2 中标注 BUFG 的输出端衰减器并在 Array1 窗口中选择 IOB 的时钟输入。松开 **Ctrl** 键，点击鼠标右键显示选项菜单并选择 **Add**。这样就最终完成了 BUFG 输出和 IOB 之间的连接，而 IOB 则用来驱动下游 ASIC 的新建接口，利用 interface\_clk90 时钟就可以正确地捕获传输的数据。

这样就完成了 ASIC 的补丁。现在重新运行设计规则 (DRC) 检查器，确保没有引入任何新的错误。在菜单条中点击 **Tools** → **DRC** → **Run**。

脚本完成并且没有错误的情况下，就可以回到菜单选择 **Tool** → **Script** → **End Recording**。这将停止并关闭脚本记录过程，下次需要这一 ASIC 补丁时你还可以重新利用这一脚本。一个不错的方法是在文本编辑器中打开脚本文件，将所有 GUI **Post** 和 **Unpost** 命令

去除。这些命令并不是必要的，而且留着它们还让脚本难于阅读。下面的文本就是我们 ASCII 补丁的脚本。如前面所述，内容非常直接，易于读取。

```
unselect -all
setattr main edit-mode Read-Write
add -s "BUFGCTRL_X0Y28" comp
clk90_bufg ;
setattr comp clk90_bufg type BUFG
unselect -all
select pin 'BUFGCTRL_X0Y28.I0'
select pin 'DCM_ADV_X0Y11.CLK90'
add
post attr net $NET_0
setattr net $NET_0 name
DCM_clk90_out
unselect -all
select pin 'OLOGIC_X0Y2.CLK'
delete
unselect -all
select pin 'ILOGIC_X0Y3.CLK'
delete
unselect -all
select pin 'ILOGIC_X0Y3.CLK'
select pin 'OLOGIC_X0Y2.CLK'
select pin 'BUFGCTRL_X0Y28.O'
add
```

```
post attr net $NET_1
setattr net $NET_1 name
clk90_bufg_out
unselect -all
drc
save -w design "patch.ncd"
"patch.pcf"
exit
end
```

仔细看看脚本内容，看能否分辨出对应 GUI 中做的动作。

很重要的一点是，可以从 GUI（在菜单条 **Tool** → **Scripts** → **Playback** 下）中或命令行回放这一脚本。要在编译脚本中加入补丁，只需要增加下面一条命令就可以了：

```
fpga_edline yourdesign.ncd
yourdesign.pcf -p yourscrip.scr
```

应当在布局布线 (PAR) 完成后再执行这一命令，即当 NCD 和 PCF 文件完成后。

FPGA Editor 真的是一项强大的用户工具，尽管并非每个人都希望或需要在设计过程中用到它。但当需要一些特殊处理或需要在某些地方突破规则以使设计功能更强时，没有其它工具能够像 FPGA Editor 一样提供如此强大的能力。你的 FAE 能够向你展示其功能，展示 FPGA Editor 如何帮助你完成设计调试和验证，当然还包括对规则的突破。🌈

*Clayton Cameron* 是赛灵思公司在多伦多市的一名资深现场应用工程师 (FAE)。他于 2000 年加入赛灵思公司，在赛灵思渥太华的办公处工作，主要为电信客户提供支持。作为一名现场应用工程师，Clayton 为客户提供支持，帮助他们解决问题。他喜欢这一工作的多样性以及每天遇到的各种挑战。业余时间，他喜欢健身，还十分享受家庭生活。