

# 第十章 FPGA 设计技巧——ISE 高级设计工具

10.1 结构向导 (Architecture Wizard)

10.2 管脚约束 (PACE)

10.3 设计约束 (Constrain Editor)

10.4 功耗分析 (Xpower)

10.5 手工布局工具 (Floorplanner)

10.6 手工布线工具 (FPGA Editor)

10.7 下载工具 (iMPACT)

10.8 PROM 文件生成 (PROM File Format)

10.9 模块化设计工具 (Modular Design)

10.9 小结

在本章中介绍 ISE 中的几个工具，这些工具在一般的设计中不是很常用，本章也仅仅对这些工具做一个简单的介绍，使得读者有一个基本的概念和印象。

## 10.1 使用 FloorPlanner 手动布局逻辑块

如果读者有电路板的设计经验，那么就很容易理解 FloorPlanner 的流程与作用了。设计电路板时候，我们首先要画出原理图，接下来将原理图转换为网表传递给布线工具，我们在布线工具中画出板框大小尺寸，决定电路板的层数，设置布线线宽，再接下来要对元器件进行布局，然后进行布线。在电路板布局布线中，可以完全手工先布局，再布线。而在 FPGA 应用的设计中，先使用语言或原理图或其他方式描述我们的设计，这类似于电路板设计中的原理图，然后进行编译、综合将设计转换为 RTL 级其他格式的网表，这类似于电路板设计中原理图设计工具向布局布线工具转换网表，再下面，我们进行映射工作，就是将设计转换成的网表与具体的目标器件中的具体资源结合起来，也即使用目标器件中的资源来实现我们设计的功能，这类似于电路板设计中将网表与具体的封装相结合，再接下来，我们可以决定使用目标 FPGA 器件中的哪些资源，这里我们就要进行布局工作了，这类似于电路板设计中的布局工作，最后，我们要将各个资源通过器件内部的互连线资源连接起来，也就是布线工作了，这类似于电路板设计中的布线工作。在一般的设计中，特别是入门阶段，由于对目标器件的内部结构及其资源不是很了解，很难对设计进行手工布局布线，我们这时候一般都是使用自动的布局布线方式。

Xilinx 软件提供了强大的工具，使得我们可以手工布局布线。如前所述，这里需要读者对目标 FPGA 器件的内部资源特别了解，这可以参阅本书前面的章节。如果不是很熟也没有关系，下面我将以极其通俗的语言来描述 FloorPlanner，写书也确实是一件很困难的事情，尤其是当前几乎所有的比较好的资料都是英文的，有些术语确实难以翻译，尽管理解它的意思，像 FloorPlanner 如果翻译成布局工具，虽然较为合适，但也难以真正表达我的理解，如果直接将英文生搬硬套地翻译过来，这是对读者不负责任的，也是应该受到谴责的，有关 FloorPlanner 等工具目前上市的中文书很少，也只有参照软件的英文说明了，有的地方也会有错误和不妥的地方，也希望读者能与我联系。讲了这些，只是希望我的这些罗罗嗦嗦的说明中，读者能轻松入门，尽早真正清楚地掌握各个工具，也希望能更快更好地发展我国的 ASIC 产业。讲了这么题外话，下面，言归正传吧。

通过前面的讲述，读者可能初步理解了 FloorPlanner 的作用，同时要说明的是，非常类似于电路板的设计流程，在 FPGA 的设计中，我们也可以将自动布局布线与手动布局布线结合起来，这有时候是一个

循环和尝试的过程，即反复进行手动布局布线进行仿真看是否有时序方面的提高，直到满足要求为止。这也是一个设计方法。

下面我们还是以具体的实例来开始学习 FloorPlanner 吧。下面以一个 2 线-4 线译码器为例来说明 FloorPlanner 的使用，这里使用的工具是 ISE6.1SP3（补丁 3）。

- 1、打开 ISE，点击 File->NewProject... 新建一个工程，输入工程名为 decoder24，工程路径为 D:\YHQProj\ISE\decoder，顶层模块类型为 HDL，如图 10-1-1 所示，点击下一步，在出现的窗口中选择目标器件为 Spartan2 xc2s100 tq144 -6，如图 10-1-2 所示，点击下一步，再点击下一步，点击下一步，点击完成；

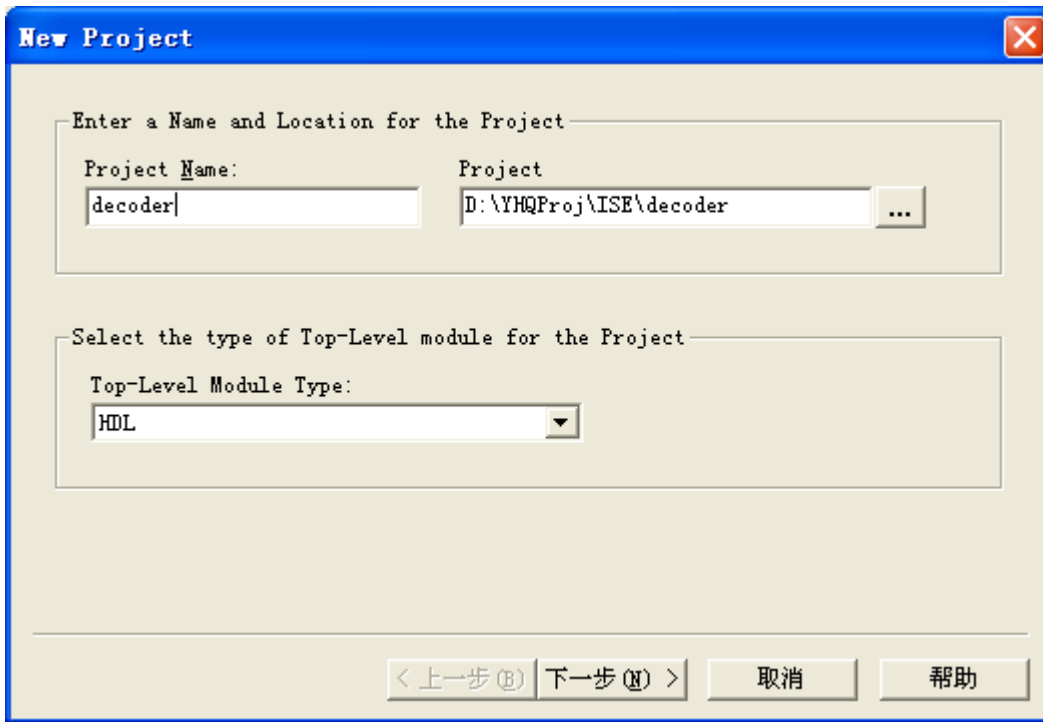


图 1 新建 ISE 工程

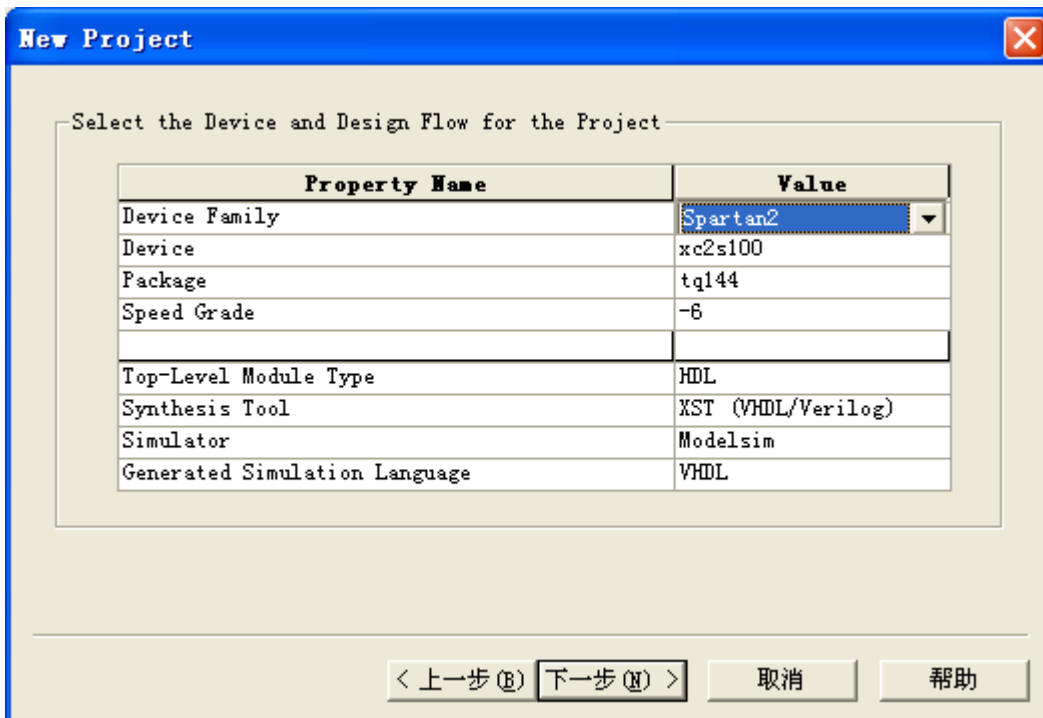


图 10-1-2 选择目标器件类型

- 2、添加源文件，点击 Project->New Source...，在出现的窗口中选择新文件类型为 VHDL Module，输入文件名为 top，如图 10-1-3 所示，点击下一步，再点击下一步，点击完成；

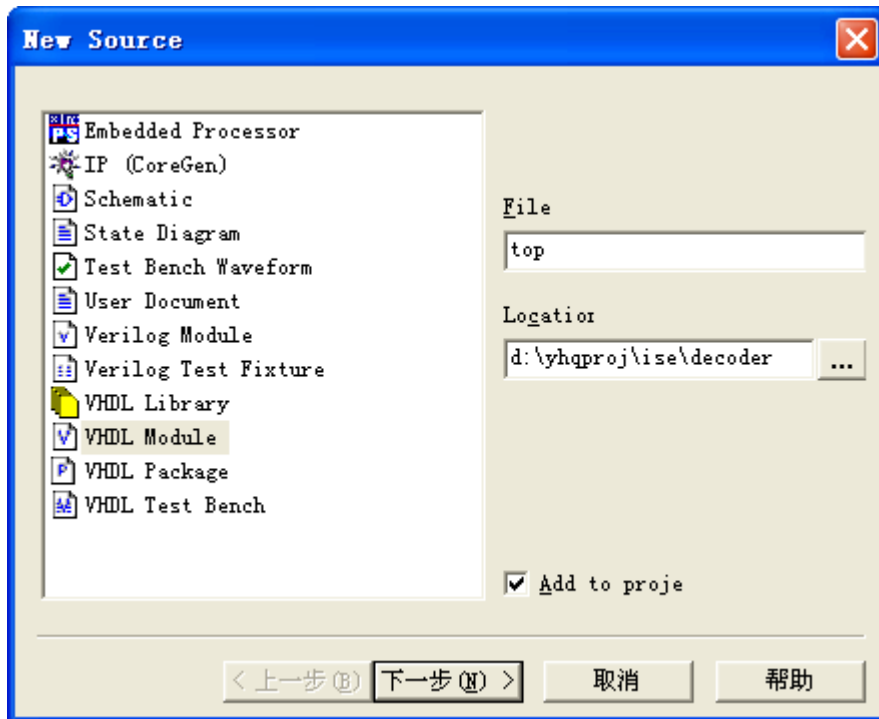


图 10-1-3 为工程添加源文件


- 3、在如图 10-1-4 的工程资源窗口 (Sources in Project) 中，双击 top-behavioral (top.vhd)，在主界面的右侧会出现该文件的内容，我们输入以下内容作为该文件的内容：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity decoder24 is
    Port ( a1,a0: in std_logic;
           y: out std_logic_vector(3 downto 0));
end decoder24;
architecture Behavioral of decoder24 is
    signal inputnum:std_logic_vector(1 downto 0);
begin
    inputnum<=a1&a0;
    process(inputnum)
    begin
        case inputnum is
            when "00"=>y<="0001";
            when "01"=>y<="0010";
            when "10"=>y<="0100";
            when "11"=>y<="1000";
            when others=>y<="XXXX";
        end case;
    end process;
end;
```

```
end process;  
end Behavioral;
```



图 10-1-4 打开设计中的源文件

- 4、 点击  保存所有文件，下面就开始综合我们的设计，看看有没有错误发生，我们在工程资源窗口（Sources in Project）中选中 top-behavioral (top.vhd)，然后在如图 10-1-5 所示的当前资源操作窗口（Processes for Source: “top-behavioral”）中，双击 Synthesize - XST，系统提示有一个警告，点击 Warnings 选项卡可以看到警告的内容为如图 10-1-6 所示的内容，因为我使用的 ISE6.1 是评估版，所以有这个警告信息；

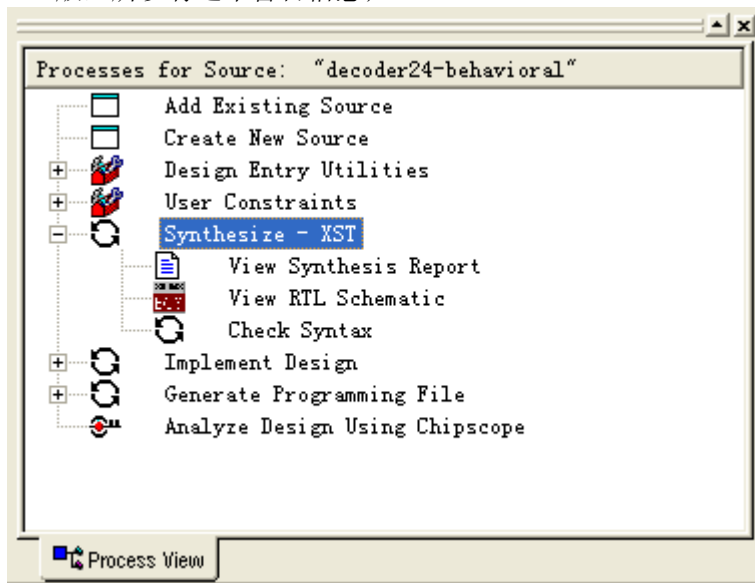


图 10-1-5 对当前工程中的资源文件进行操作

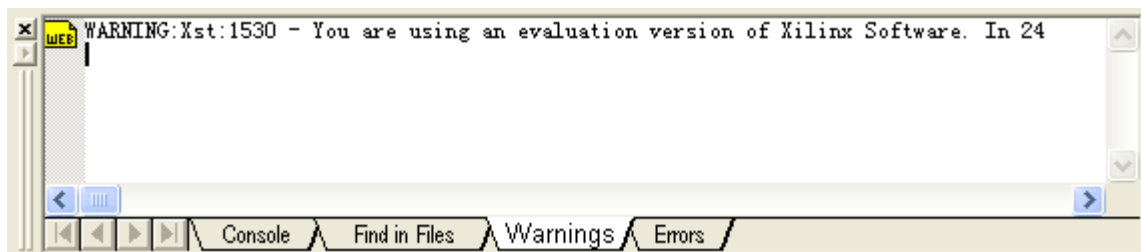



图 10-1-6 点击 Warnings 查看警告信息

- 5、 这里为了后面的理解，我们先看看综合工具综合出的 RTL 视图是什么样的，我们双击 View RTL Schematic，这是会出现一个如图 10-1-7 所示的窗口，我们在如图 10-1-8 所示的 RTL Design Hierarchy 窗口中选中 Mdecod\_y，然后点击图标 ，这时我们可以在右下部分的界面中看到如图

10-1-9 所示的 RTL 原理图，我们双击图中最下面的方框可以看到如图 10-1-10 所示的底层 RTL 原理图，由此可以看出，设计由四个二输入与门和三个非门组成，关闭 Xilinx ECS 窗口；

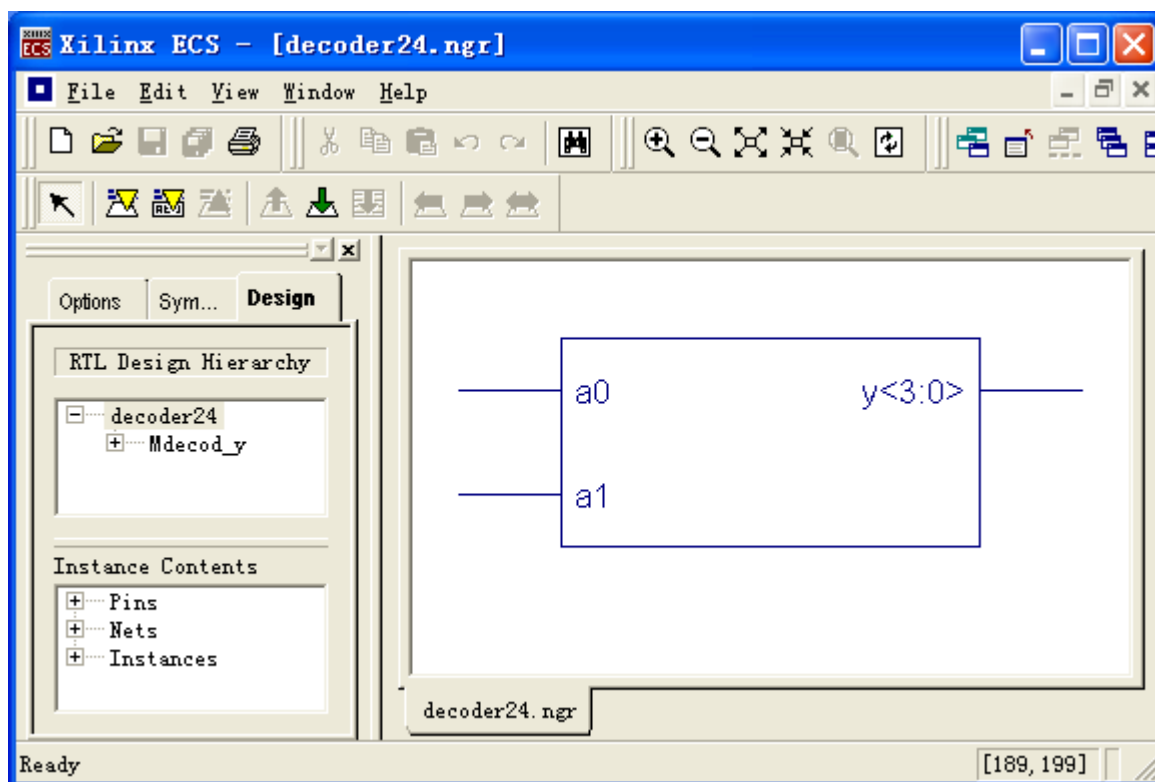


图 10-1-7 Xilinx ECS 窗口

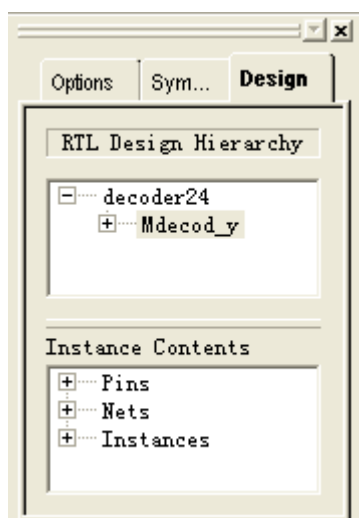


图 10-1-8 RTL 设计层次窗口

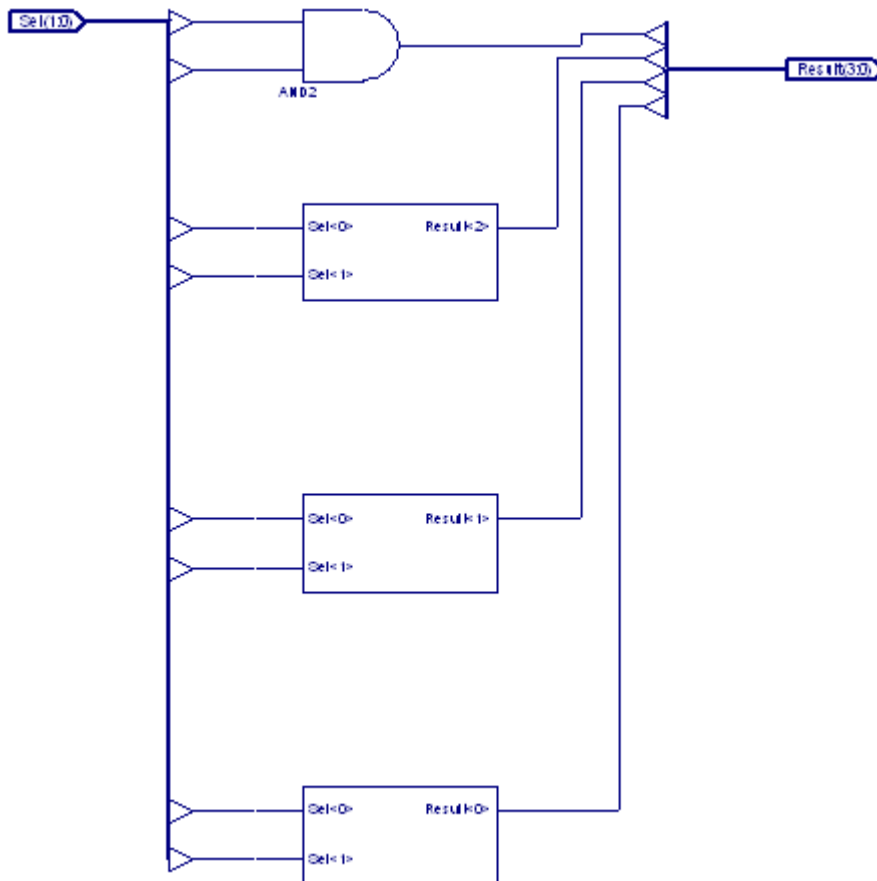


图 10-1-9 综合成的 RTL 原理图

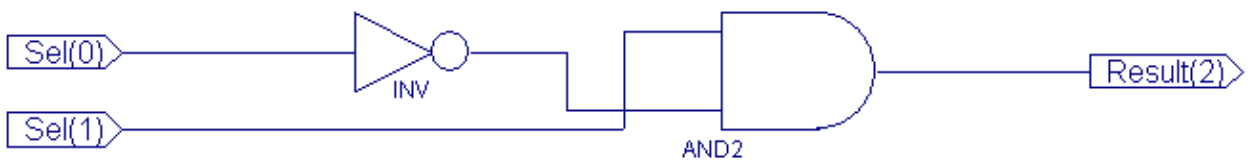


图 10-1-10 底层 RTL 原理图

- 我们双击如图 10-1-11 所示的 Floorplan Design，这时会出现如图 10-1-12 所示的窗口，这就是 Floorplanner 的主界面；

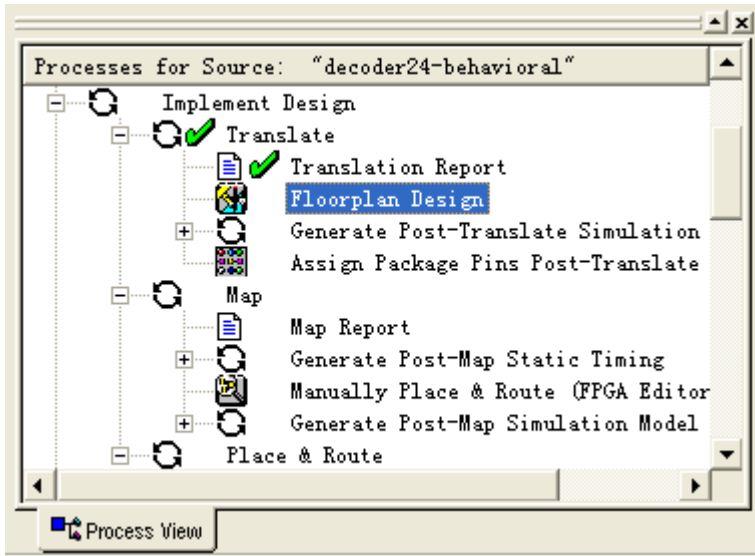


图 10-1-11 对设计中的文件进行操作

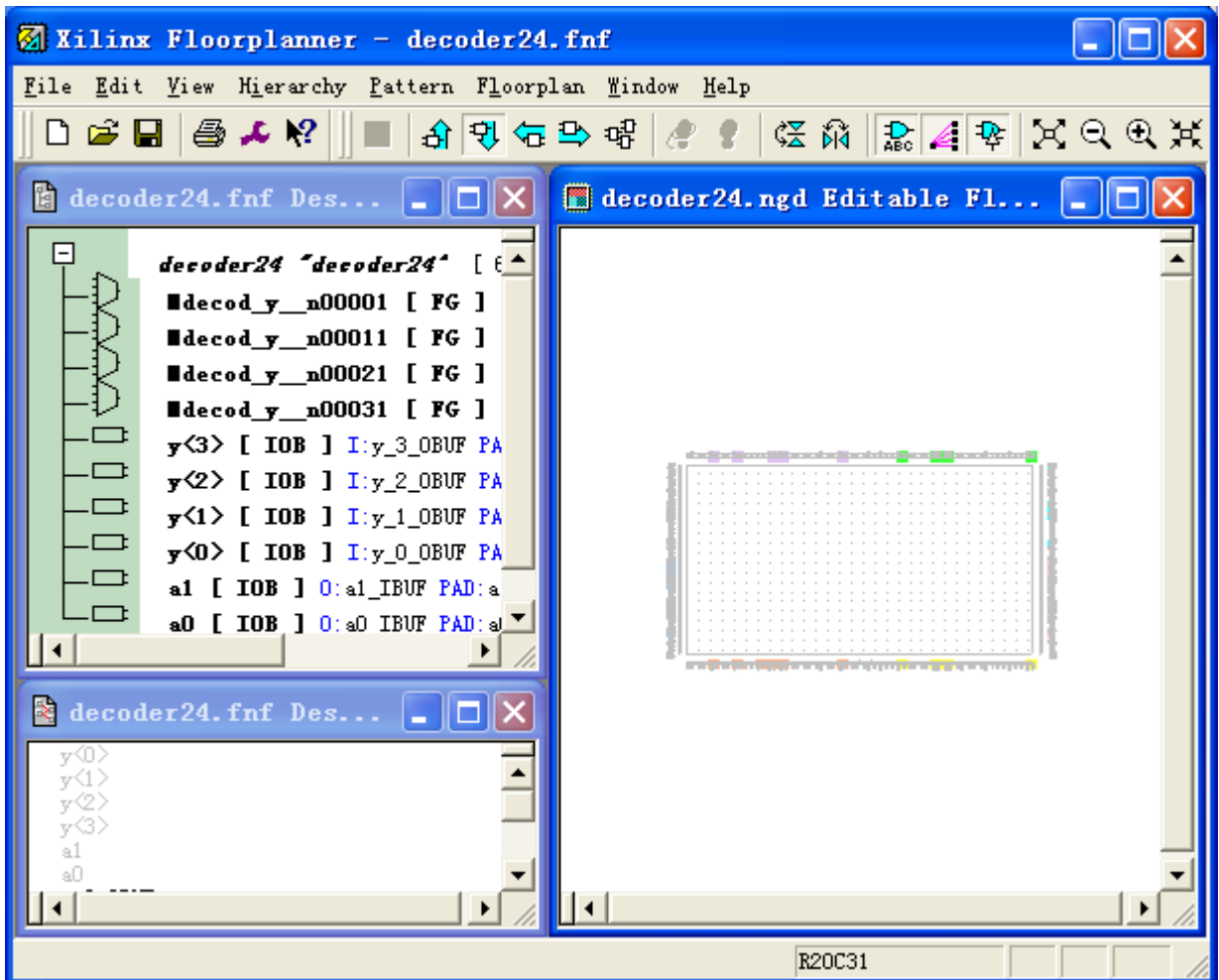



图 10-1-12 Xilinx Floorplanner 窗口

- 7、在 Floorplanner 主界面中，点击  将图放大到足够大，这时候就可以看到我们所选取的 FPGA 的内部资源图，如图 10-1-13 所示，下面我们所要进行的工作是将各个模块进行布局，类似于电路板

的布局。我们点击 Windows->Arrange Default，使得在主界面中同时显示设计层次以及 FPGA 内部资源窗口，我们在如图 10-1-14 所示的层次窗口中，选中最上面一个逻辑块，将它用鼠标拖拉到 FPGA 内部资源窗口中，在合适的位置放下来，注意拖拉的时候不要在层次窗口中向下拖动，而是横向拖到资源窗口中，在没有禁止符号的地方放置下来，同理放置层次窗口中下面的各个逻辑块，放置完毕后的资源窗口如图 10-1-15 所示，如果将图放足够大，可以看到每一个逻辑块的编号；

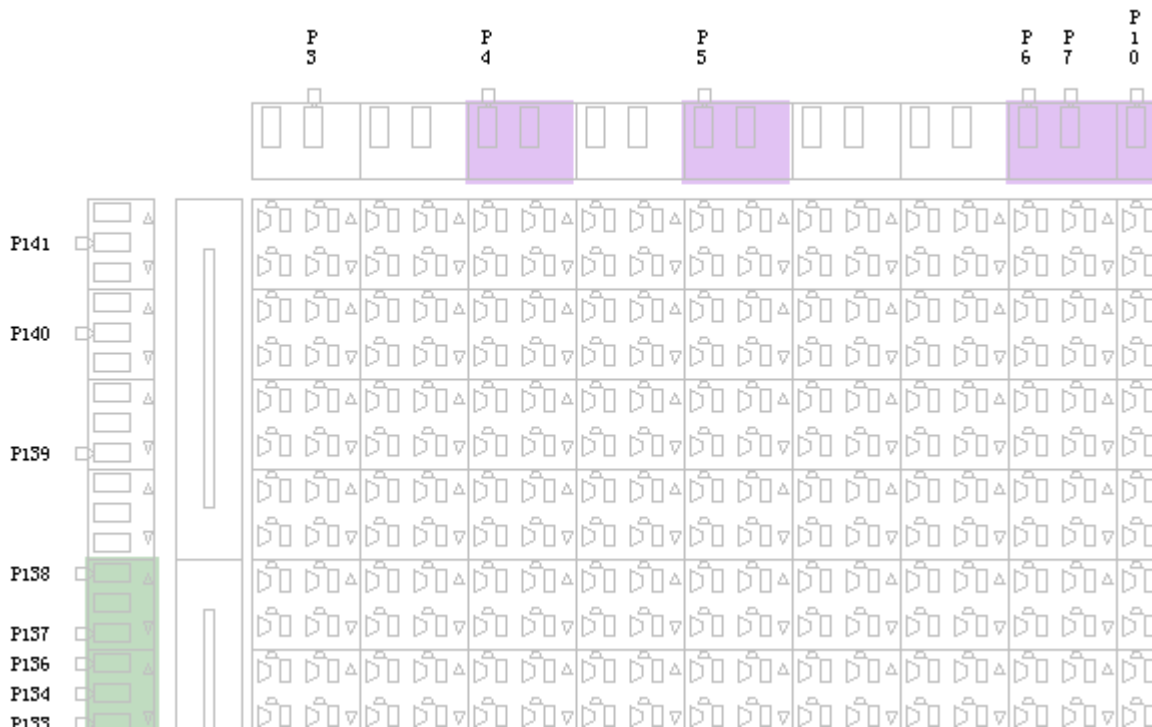


图 10-1-13 选定的 FPGA 内部资源图

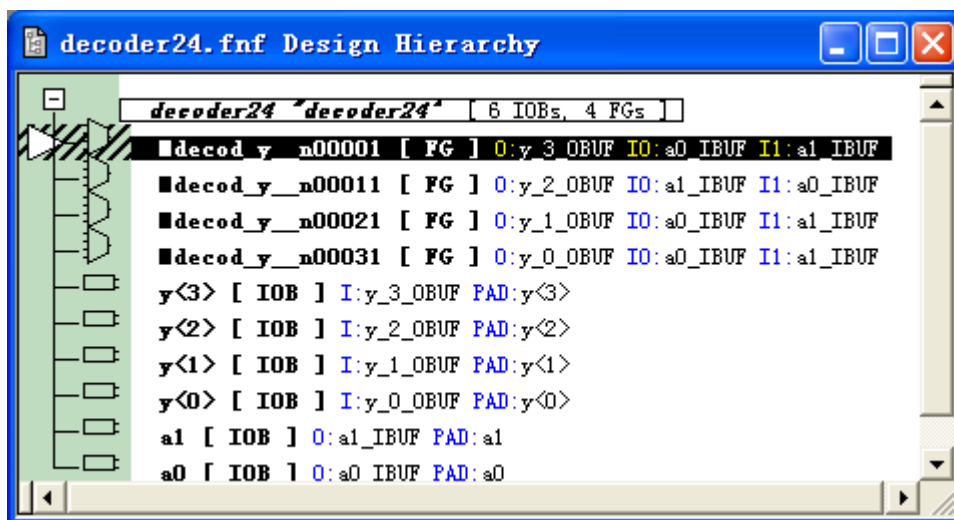


图 10-1-14 设计层次窗口



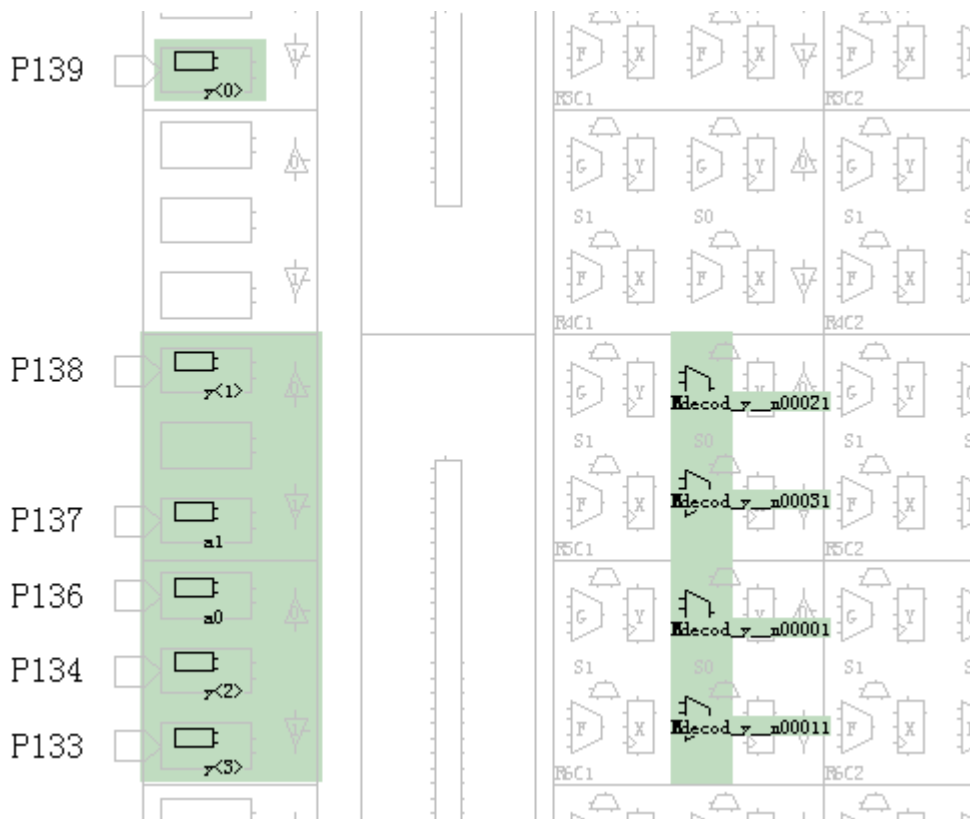



图 10-1-15 放置完毕后的资源窗口

- 8、 点击  保存设计，这时候会出现窗口提示输入文件名字，需保存两个文件，一个是 Floorplanner 的网表文件，一个是约束文件，在约束文件中有刚刚我们所设置的管脚绑定等信息；
- 9、 退出 Floorplanner 窗口；
- 10、 为设计添加刚刚生产的约束文件，我们点击 Project->Add Source...，选择刚刚生成的 decoder24.ucf 文件，点击打开将其添加到工程中；
- 11、 我们在工程资源窗口 (Sources in Project) 中选中 top-behavioral (top.vhd)，然后在如图 10-1-16 所示的当前资源操作窗口 (Processes for Source: "top-behavioral") 中，双击 View/Edit Placed Design(FPGA Editor)，可以看到布线后的内部资源窗口如图 10-1-17 所示，当然我们也可以手工布线，这将在后续章节讲述；

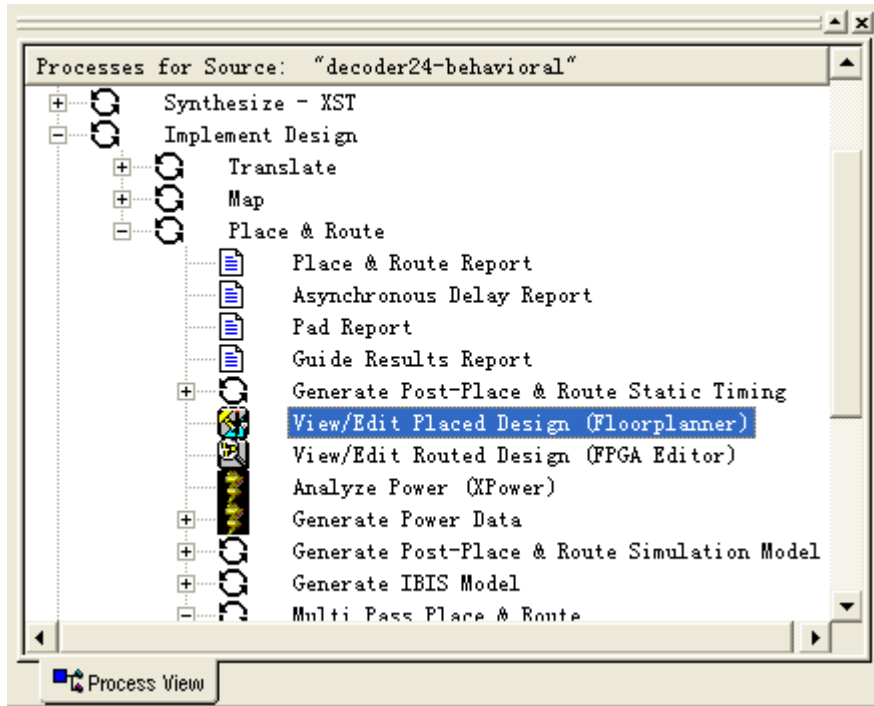


图 10-1-16 当前资源操作窗口

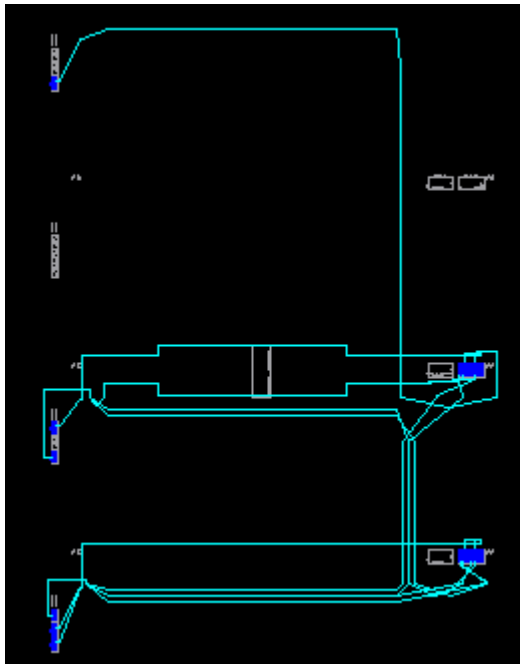


图 10-1-17 布线后的资源图

12、至此，我们已经初步了解了 Floorplanner 的基本操作，这是最基本的操作，有关的更深入的操作还是要读者去慢慢摸索了。其实 Floorplanner 的主要功能是将设计中的逻辑最佳地分组，同时也可以手动进行放置逻辑块的操作。因为例子很简单没有涉及分组操作，分组就是将联系紧密连线较多的逻辑块合成一个组，以后就可以按组进行放置逻辑块了。

## 10.2 使用 FPGA Editor 进行手动布局布线

如上节开始所述，在 ISE 中提供了手动布局布线的功能，这些功能在设计很大的时候可以提高设计的时序性能以及能使设计的密度更大更容易布线成功等等，这些功能是通过 Floorplanner 和 FPGA Editor 实现的，同时，我们也可以利用 Floorplanner 和 FPGA Editor 生成约束文件。在本节中，我们讨论手动布线工具 FPGA Editor，我们还是通过一个例子来简单地说明其使用方法。

- 1、打开上一节设计的工程，可以通过双击工程文件或启动 ISE 后打开工程文件来打开它；
- 2、在如图 10-2-1 的窗口中，选中 decoder24-behavioral(top.vhd)，在如图 10-2-2 所示的窗口中双击 Manually Place & Route (FPGA Editor)，这时候会出现如图 10-2-3 所示的窗口，我们可以一直放大该窗口，发现这完全是目标 FGPA 器件的内部逻辑块图，还没有引入我们的设计；

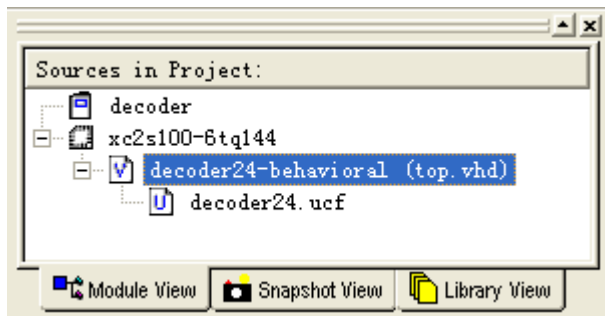


图 10-2-1 工程中的资源窗口

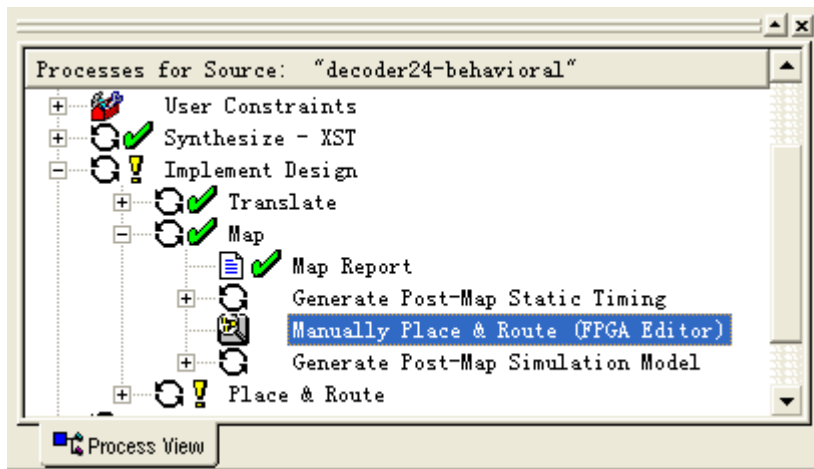


图 10-2-2 当前文件的可用进程

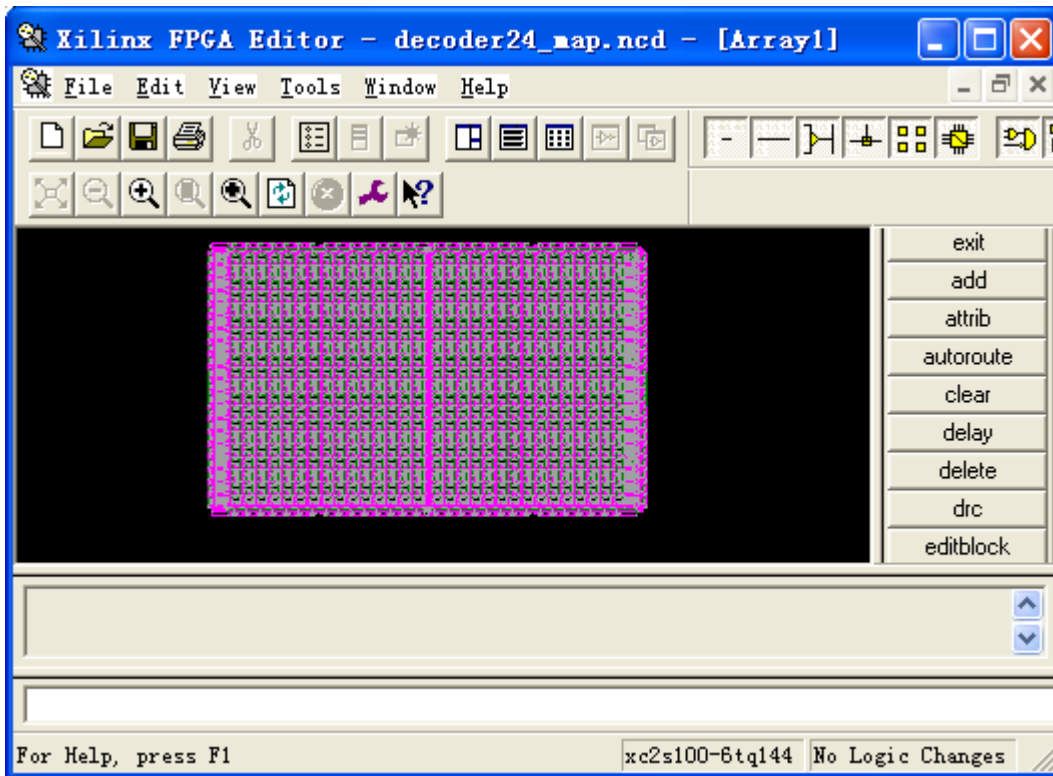



图 10-2-3 FPGA Editor 窗口

- 3、 下面就要引入我们的设计，点击图标，会出现一个打开窗口，提示输入我们的设计以及物理约束文件，所谓的物理约束文件一般是在映射过程中创建的，或者是我们通过 FPGA Editor 创建的，我们输入设计以及约束文件如图 10-2-4 所示，在 Edit Mode 中选择 Read Write 以便可以对设计中的逻辑块位置重新调整，点击 OK；

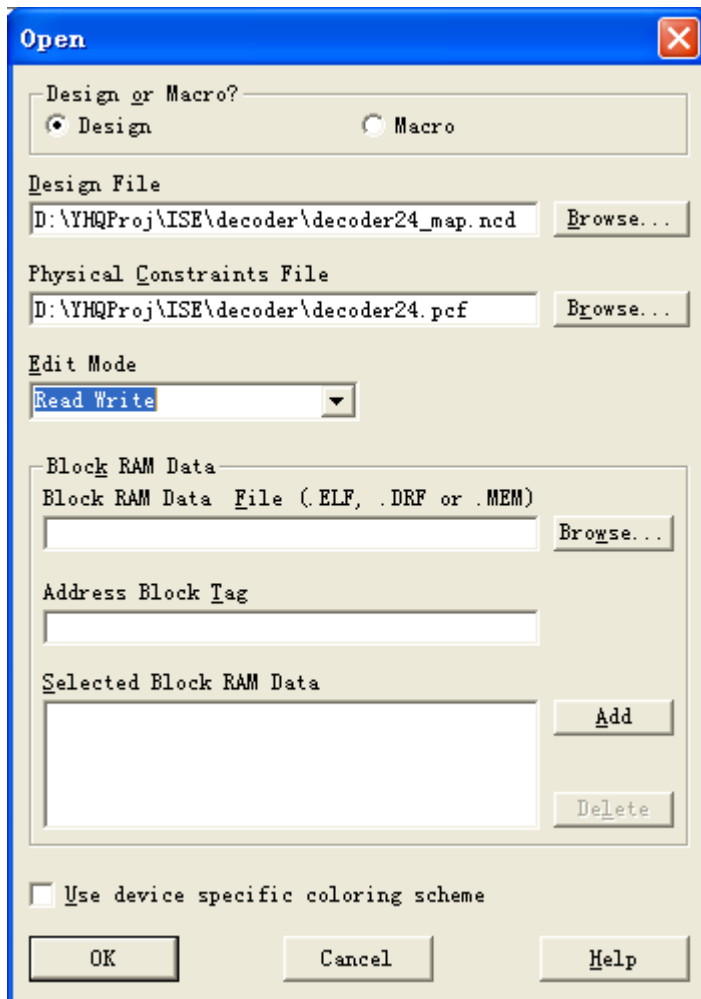


图 10-2-4 为 FPGA Editor 添加设计文件以及物理约束文件

4、 有时候，这一步会有警告信息出现，警告信息如图 10-2-5 所示，我们点击是 (Y)；

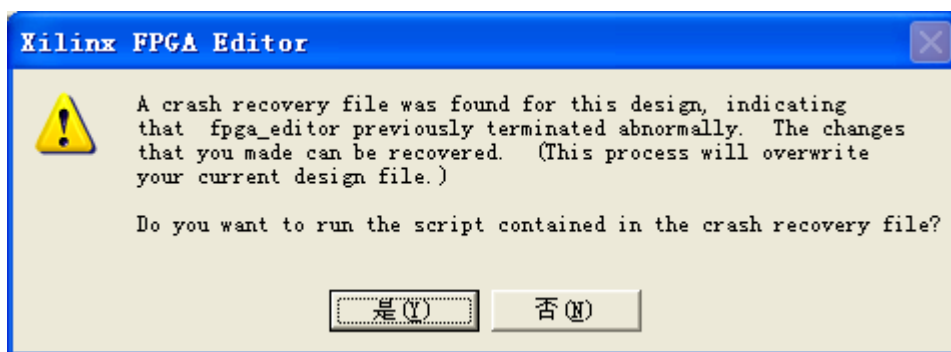


图 10-2-5 警告信息

5、 有时候这一步又会有错误信息，如图 10-2-6 所示，我们点击确定，好了，这时候总算看到我们上一节布局的结果了，如图 10-2-7 所示，这个界面就是我们进行布线操作的界面；

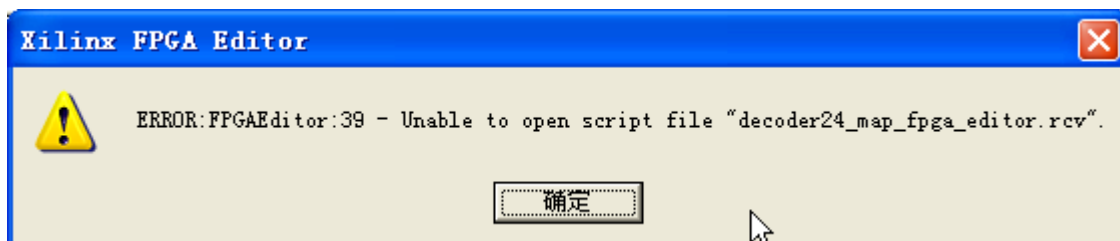


图 10-2-6 错误信息

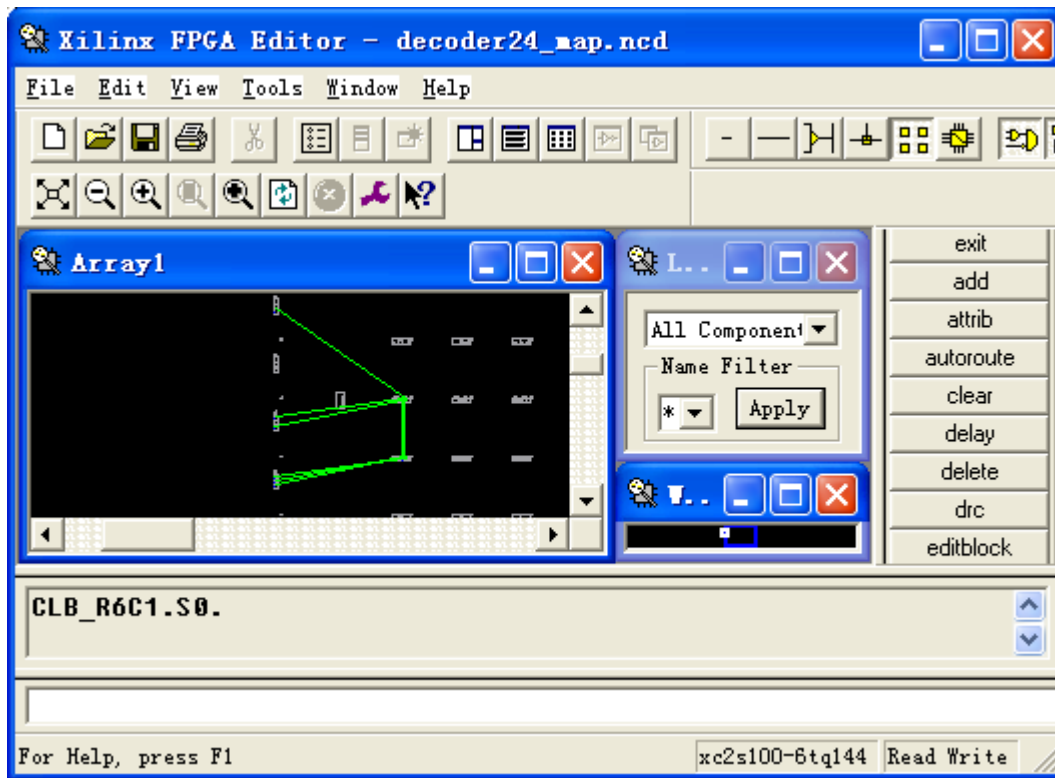


图 10-2-7 FPGA Editor 打开了布局结果后的界面

6、 我们先试一下自动布线，点击 Tools->Route->Auto Route All，这时候，可以看到自动布线后的结果如图 10-2-8 所示；

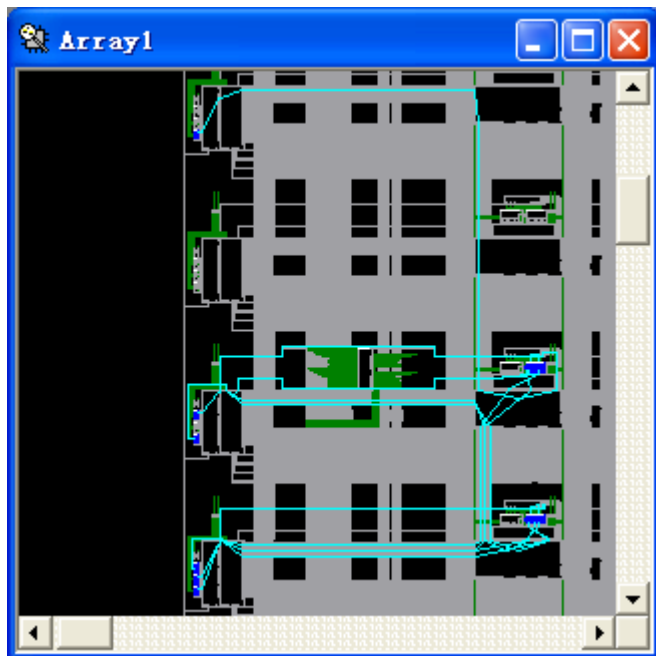





图 10-2-8 自动布局布线结果

7、 再试一下手动布线，先把自动布线结果删除，点击 Tools->Route->Unroute All，在出现的确认对话框中点击确定，这时候又回到了布线前的状态，我们可以把图放大，点击 y<1>中有鼠线的引脚，这里鼠线就是图中的绿色的线，表示两引脚之间有连接，但还没有布线，布线后，鼠线会消失，如果放大图后没有显示标签 y<1>，可以点击 **A** 来显示，另外要注意的就是这里一定要选择引脚，

而不能选择鼠线，选中以后，再按 **Ctrl** 键的同时点击与该鼠线相连的另一个管脚，选中后，会在该管脚上有一个红色的三角表示选中，选中两端之后，点击界面右侧的按钮 ，这时候，可以看到这条鼠线已布线，这就是手动布线，也许你感觉点击 **route** 仍是自动布线，两点间的连线没有办法，自动布线也许与手动布线结果相同，但对于一个连接多个逻辑引脚的鼠线来说就有一个布线顺序的问题，这个布线顺序就体现了手动功能，我们依次选中各个相连的逻辑引脚，再按  进行布线，可以发现布线后的连接顺序是刚刚选取的顺序，我们可以对图中的 **a1** 进行布线，**a1** 共有 5 个逻辑引脚相连，我们可以选择不同的顺序试一下，一个技巧是先选中鼠线，以便查看哪些逻辑引脚相连，在点击 **route** 前再点一下该鼠线以不选中，再点击  即可，图 10-2-9 显示了一个手工布线的结果，在其中有部分连线没有布线，在实际设计中也可以只布线几个关键的逻辑引脚，其余的自动布线完成；

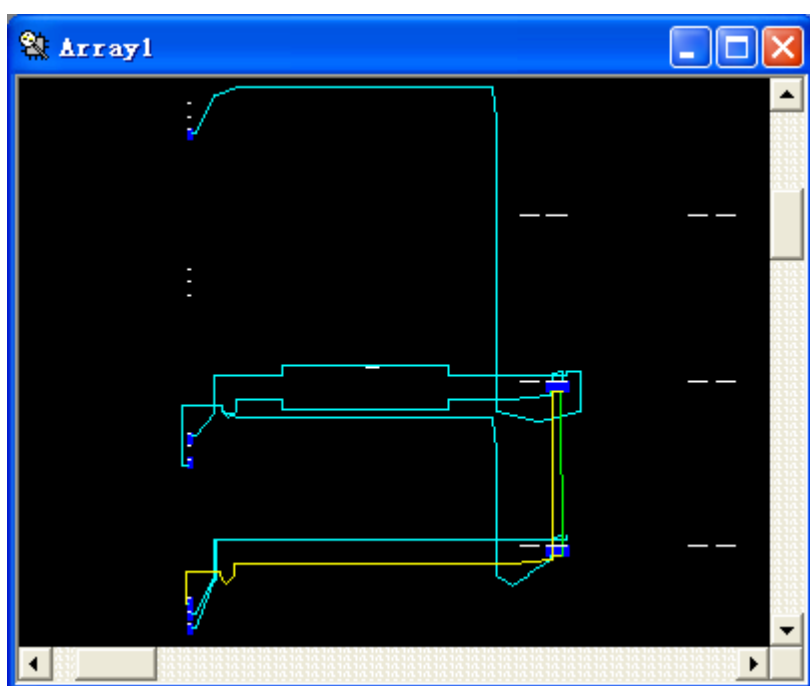


图 10-2-9 手工布线结果

- 手工布线结束后要进行规则检查，看看布线结果是否符合要求，点击 **Tools->DRC->Setup...** 会出现设置检查规则的窗口，如图 10-2-10 所示，点击 **Tools->DRC->Run**，看看有没有错误和警告信息，如果有错误需要更改相关的地方；

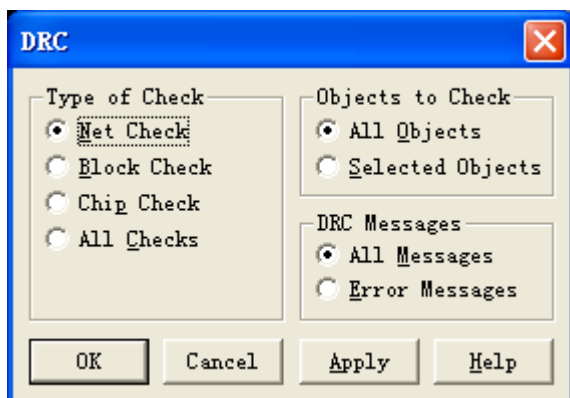



图 10-2-10 设置 DRC 检查检查

9、 点击图标保存设计并关闭 FPGA Editor;

10、 双击如图 10-2-11 所示窗口中的 View/Edit Routed Design (FPGA Editor), 可以看到如图 10-2-12 所示的最终布线结果;

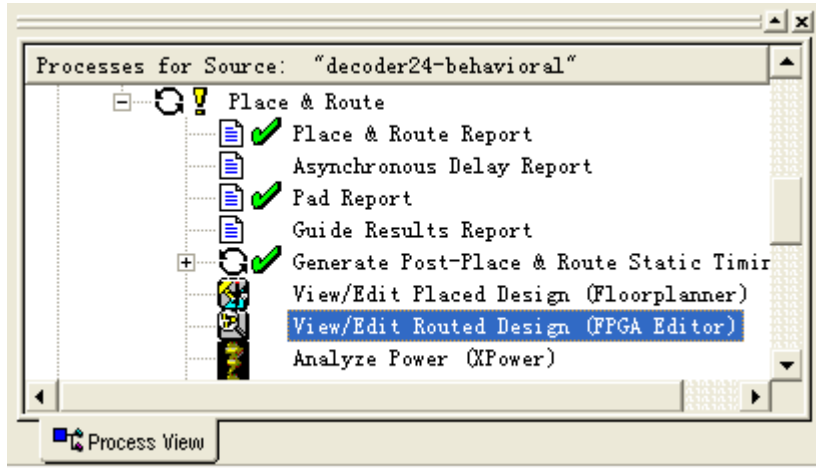


图 10-2-11 当前文件的可用进程

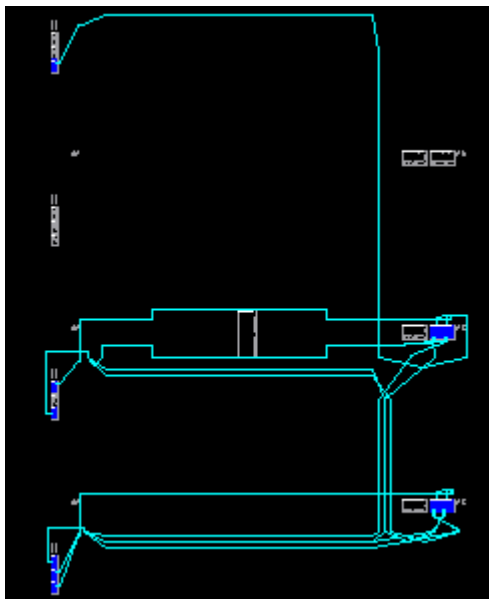


图 10-2-12 布线最终结果

在本节仅仅简略的讲述了 FPGA Editor 的操作流程, 还有许多其他的方面没有介绍, 本节仅仅起到入门的作用。若读者需要更深入地使用该软件还要自己多多练习多多尝试了。

### 10.3 使用 XPower 分析设计的功耗

Xpower 是 ISE 集成的一个功耗分析工具, 使用该工具可以分析使用的目标器件的功耗, 并能分析出电池的使用时间等信息, 关于功耗分析方面的理论知识及计算公式, 读者可以参阅其他教材或帮助文档, 在本节仅仅就软件本身的使用做一个简略的介绍。本节的生成 VCD 文件部分需要安装 ModelSim 软件, 如



如果没有安装 ModelSim 读者可以略过生成 VCD 文件的几个步骤。

下面还是以我们上面设计的例子来说明。

- 1、使用 ISE 打开上一节的设计；
- 2、新建一个仿真波形输入文件，点击 Project->New Source...，会出现一个窗口提示我们输入新建资源的名称及类型，我们选择类型为 Test Bench Waveform，文件名为 top\_tbw，如图 10-3-1 所示，点击“下一步”，出现选择与其关联的文件，我们选择 decoder24，其实我们的设计中也只有这一个 VHDL 源文件，点击“下一步”，出现界面给出我们新建文件的信息，点击“完成”；

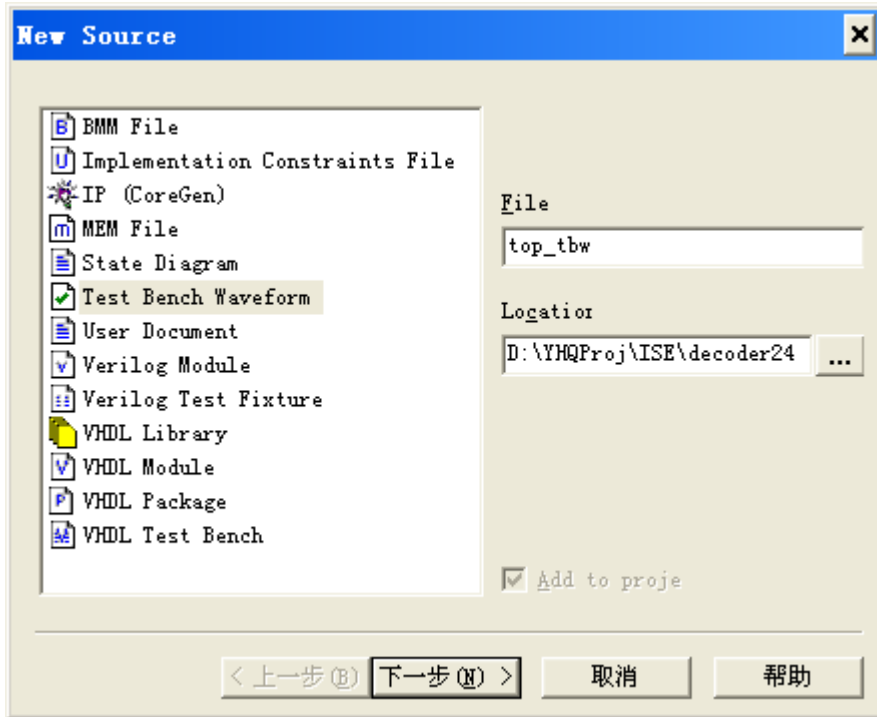



图 10-3-1

- 3、这时候出现了初试化时序的设置，我们直接点击“OK”，这时会在 ISE 内出现了设置波形的窗口，我们设置波形如图 10-3-2 所示，点击图中的  保存；

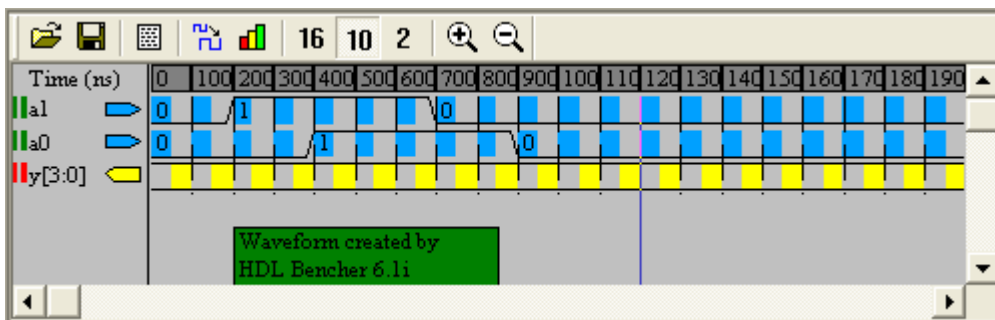


图 10-3-2 设置波形

- 4、我们设置仿真波形的目的是得到仿真中生产的 VCD 文件，为了得到 VCD 文件，我们需要进行一些设置，首先在 Sources in Project 窗口中选中 top\_tbw(top\_tbw.tbw)，然后在 Process for Source: “top\_tbw”窗口中选中 Simulate Post-Place & Route VHDL Model，然后点击右键，选择 Properties...，这时候会出现如图 10-3-3 所示的窗口，我们在 Generate VCD File 后面打勾表示在仿真时生成 VCD 文件，点击确定；

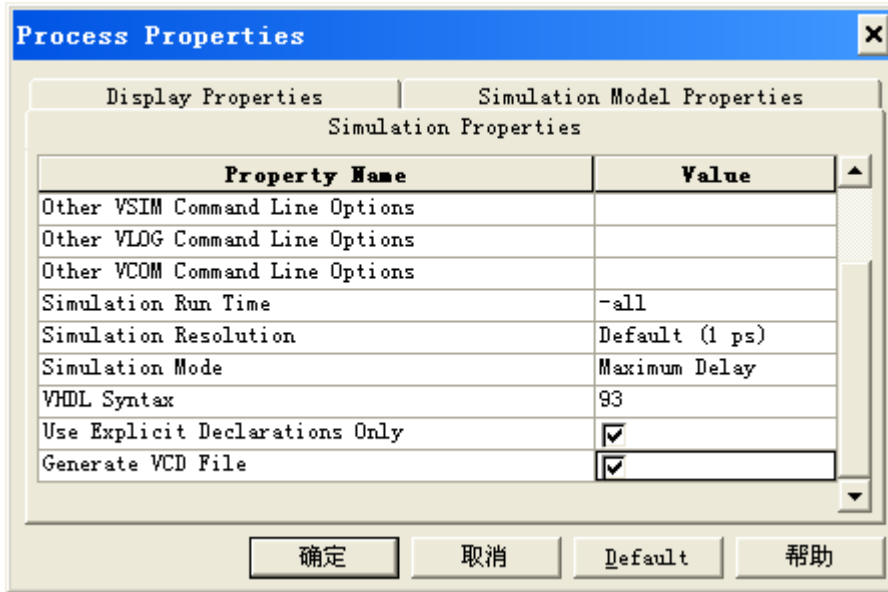


图 10-3-3 设置仿真选项

- 5、 接下来我们就要运行仿真以生成 VCD 文件，方法是双击 Simulate Post-Place & Route VHDL Model，这时候会自动运行 ModelSim，并打开了仿真波形如图 10-3-4 所示，注意这里的仿真需要一些 Xilinx 提供的仿真库，在 ModelSimXE 版本中包括了这些库，如果使用 ModelSimSE 等版本需要首先编译这些库文件，我们退出 ModelSim；

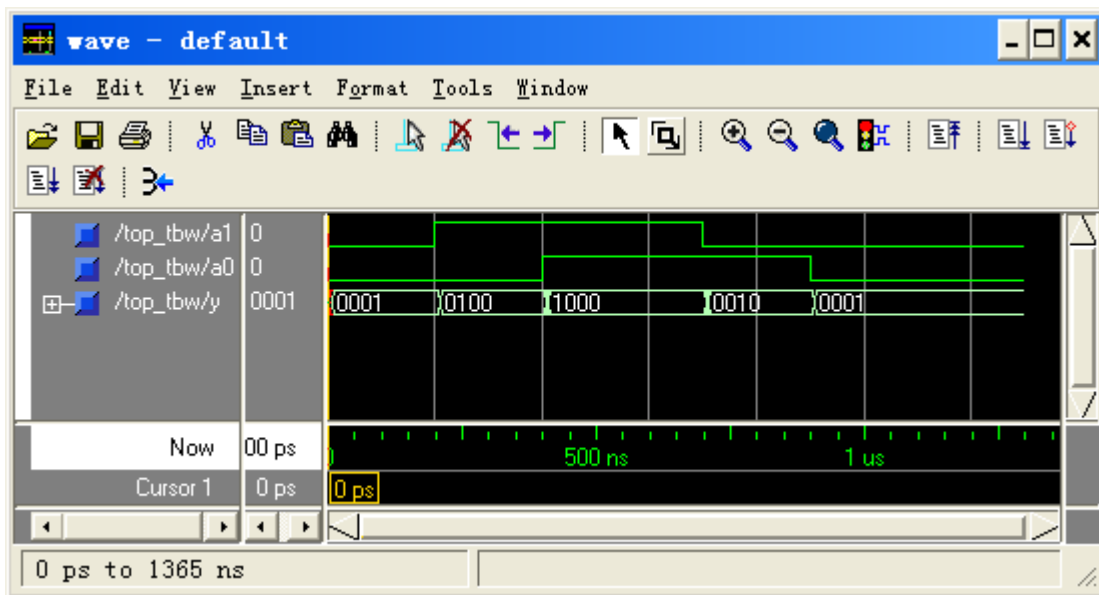


图 10-3-4 使用 ModelSim 进行仿真的波形

- 6、 接下来就要进行功耗分析了，首先在 Sources in Project 窗口中选中 decoder24-behavioral(top.vhd)，然后在 Process for Source: “top\_tbw” 窗口中双击 Analyze Power(XPower)，如图 10-3-5 所示，这时候会出现 XPower 主界面，

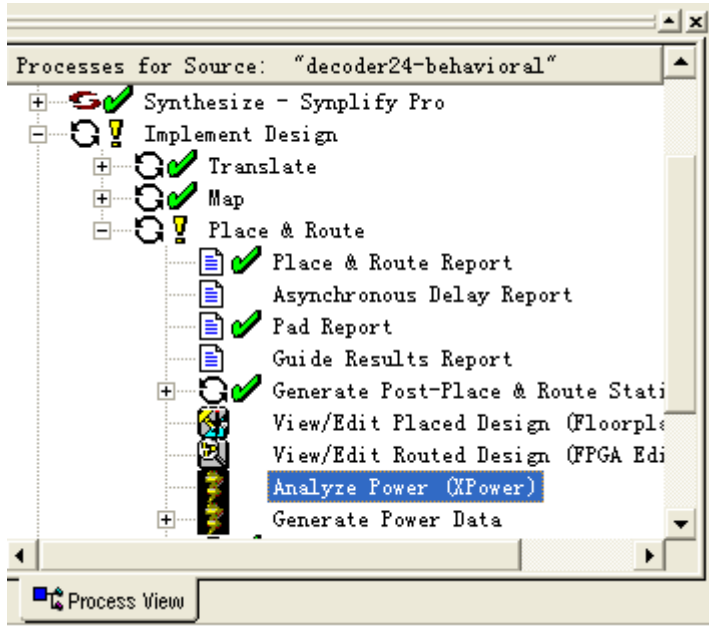


图 10-3-5

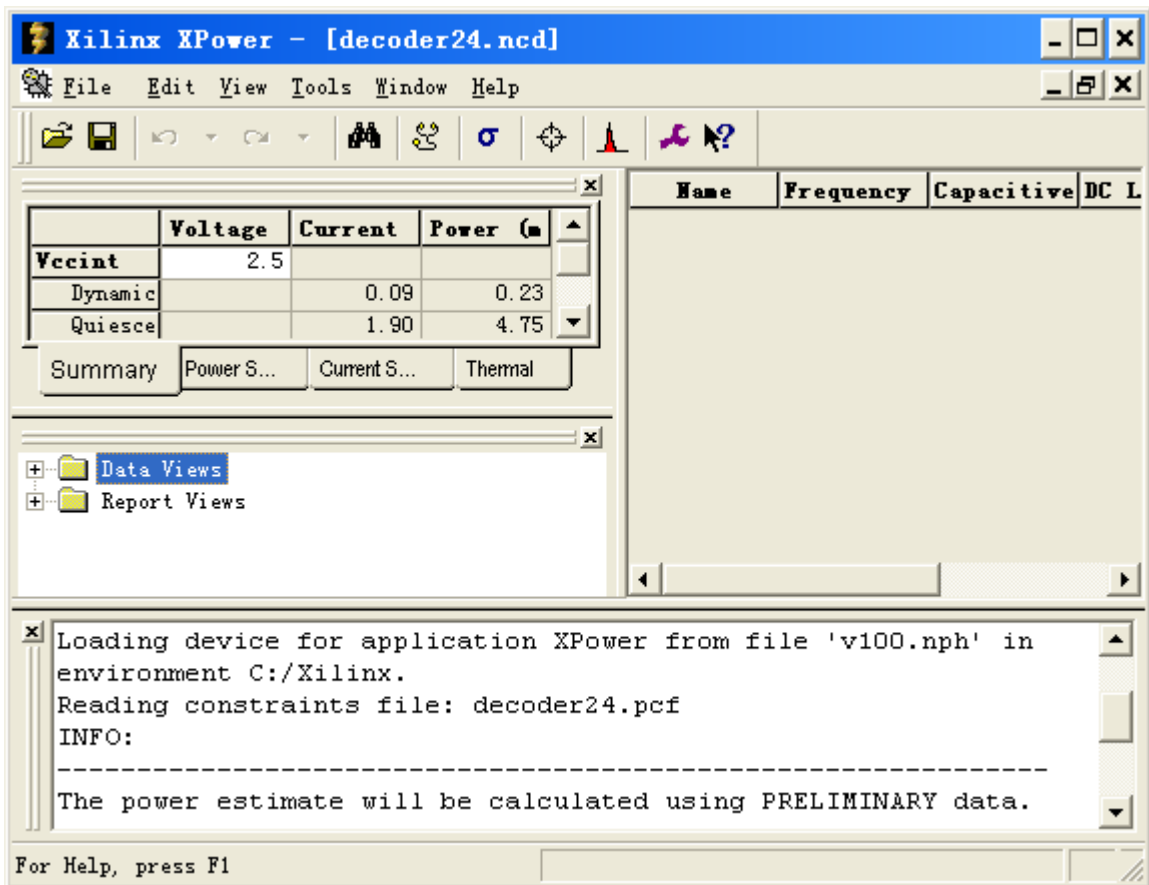


图 10-3-6 XPower 主界面

- 7、我们点击图中左下脚 Report Views 前面的加号，再点击 Power Report，这时候，会在界面的右侧出现功耗报告，报告的内容有功耗分析、温度分析以及建议的解耦电容参数等信息，本例的报告如下：

```

-----
Release 6.1.03i - XPower SoftwareVersion:G.26
Copyright (c) 1995-2003 Xilinx, Inc. All rights reserved.
Design:      decoder24
Preferences: decoder24.pcf
VCD File:    D:\YHQProj\ISE\decoder24\decoder24.vcd
Part:        2s100tq144-6
Data version: PRELIMINARY,v1.0,07-31-02

```

```

Power summary:
-----
Total estimated power consumption:      15
Peak Power consumption:                14351
-----
          ---
          Vccint 2.50V:      2      5
          Vcco33 3.30V:      3     10
          ---
          Inputs:           0      0
          Logic:            0      0
          Outputs:
          Vcco33            1      3
          Signals:         0      0
          ---
          Quiescent Vccint 2.50V:      2      5
          Quiescent Vcco33 3.30V:      2      7

```

```

Thermal summary:
-----
Estimated junction temperature:      25C
Ambient temp: 25C
Case temp: 25C
Theta J-A range: 31 - 32C/W

```

```

Decoupling Network Summary:
-----
Cap Range (uF)  #
-----
Capacitor Recommendations:
Total for Vccint :      8
          470.0 - 1000.0 : 1
          0.0470 - 0.2200 : 1
          0.0100 - 0.0470 : 2
          0.0010 - 0.0047 : 4
          ---
Total for Vcco33 :      3
          470.0 - 1000.0 : 1

```

Analysis completed: Thu Feb 12 10:52:23 2004

- 8、查看电池使用时间，在主界面的左上边，我们可以看到如图 10-3-7 所示的窗口，在其中可以设置电池的容量等参数，如图显示，电池寿命为 565.57 小时；

	Voltage (V)	Current (mA)	Power (mW)
<b>Vccint</b>	2.5		
Dynamic		0.09	0.23
Quiescent		1.90	4.75
<b>Vcco33</b>	3.3		
Dynamic		1.04	3.45
Quiescent		2.00	6.60
<b>Total Power</b>			15.03
Peak Power			14350.86
Startup Current		500.00	
Battery Capacity (mA Hours)			2850.00
Battery Life (Hours)			565.57
Confidence Level			Accurate

Summary Power Subtotals Current Subtotals Thermal

图 10-3-7 查看电池使用时间

- 9、如果发现设计的功耗不满足要求，可以通过更改布局布线、约束时序甚至重新设计等方法来改进设计。
- 10、刚刚生成报告所用的参数有的是默认值有的来自于 VCD 文件，我们也可以自己设置这些参数，在图 10-3-7 中点击不同的选项卡，可以对底色为白的表格进行设置，在该图中主要设置一些外部的和总体的参数，如内核电压 (Vccint)、IO 电压 (Vcco33)、电池容量 (Battery Life)、环境温度 (Ambient Temperature)、器件级别 (Part Type, 如商业级、工业级、军品级)、气流 (Airflow)、封装 (Package) 等信息，而另外一些 FPGA 内部的各个信号的信息则通过点击主界面左下脚的 DataViews 中来选择每一个端口和逻辑来设置，当我们选中输入、输出或逻辑块时，在主界面的右侧出现一些表格，可以对其参数作一些设置。对于输入可以设置其缓冲器的信号翻转频率进行设置，对于输出，可以设置输出缓冲器的容性负载值 (Capacitive Load)、载入电流 (DC Load) 进行设置。在 VCD 文件中保存有这些信息，因此读入 VCD 文件就可以不用进行设置内部一些参数了。