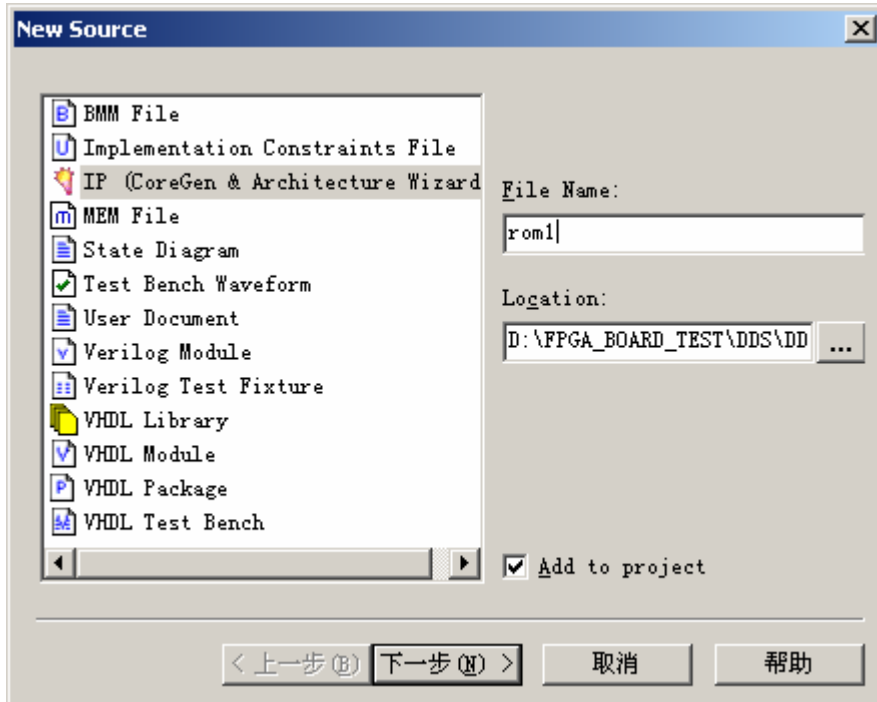


XILINX ROM 使用中文教程

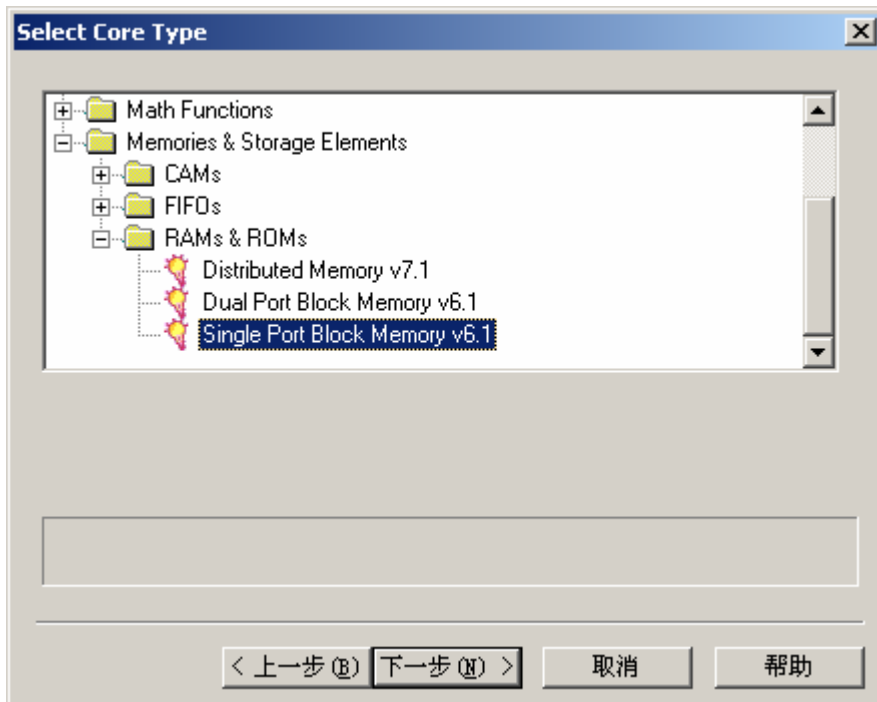
本文紧供参考学习，不得用于商业用途
适合 XILINX 的 CPLD/FPGA 用户

FPGA 内部有 BLOCK RAM 和分布式 RAM,这些资源都可以作为 ROM(ROM 只是有初始值的 RAM,而且该初始值不能改变)。有些低端的 FPGA 可能没有 BLOCK RAM.本文主要介绍 ROM 的使用方法，RAM 的使用方法类同。

打开 XILINX 的 ISE, NEW SOURCE 中选择 IP，写好 NAME，点击下一步。



在下图中选择你要的 IP 核。

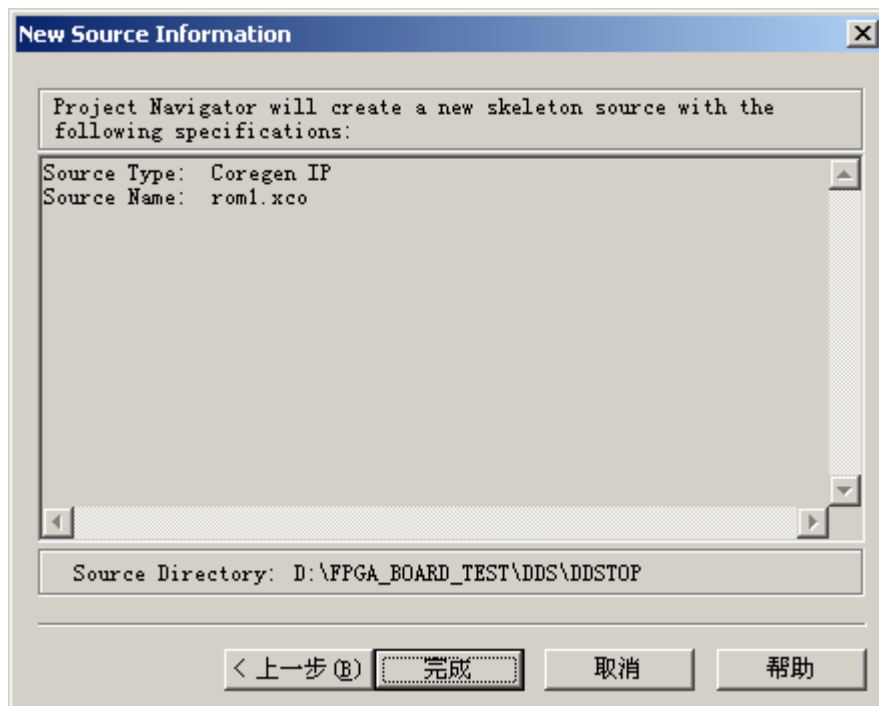


第一个是使用分布式 ROM——如果你的 ROM 不是很大的话，而且内部 BLOCK RAM 资源有限的情况下，使用分布式 ROM 可以为你节约不少 BLOCK RAM,但是它会占用一些逻辑资源。

第二个是双口 BLOCK RAM。

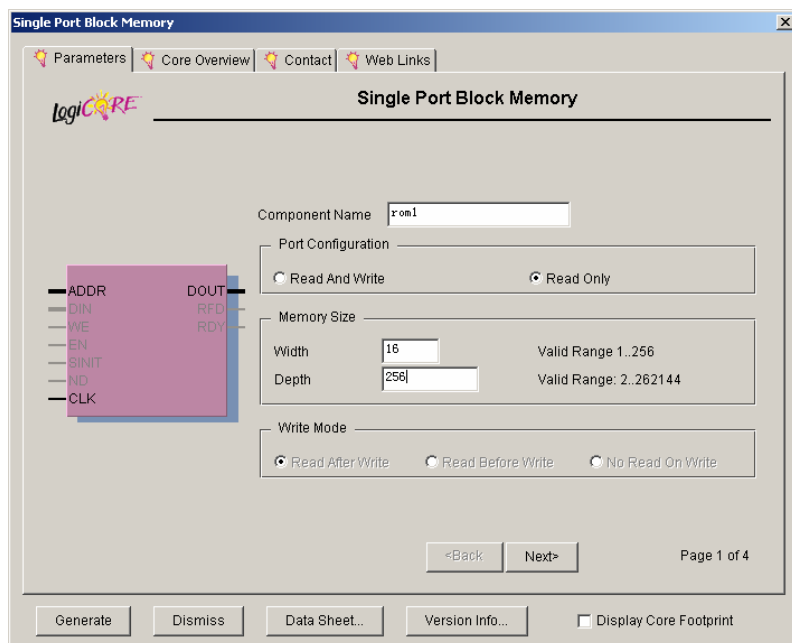
第三个是单口 BLOCK RAM。

选择你想要使用的 IP 核，（以常用的单口 BLOCK RAM 为例）点击 NEXT



点击完成。

进入如下图界面。

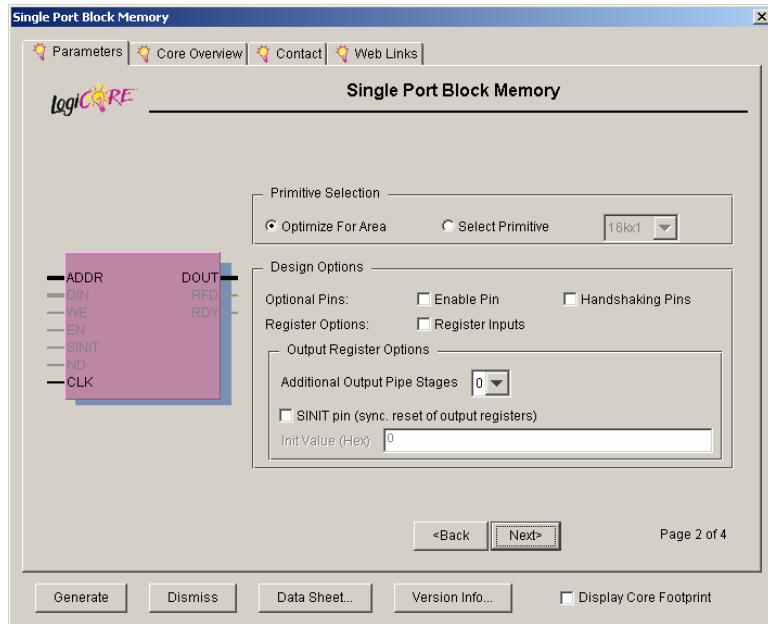


在 component name 内填写模块名（要小写）

第二项选择 READ ONLY（选 READ AND WRITE 就是 RAM），表示要生成只读的 ROM。

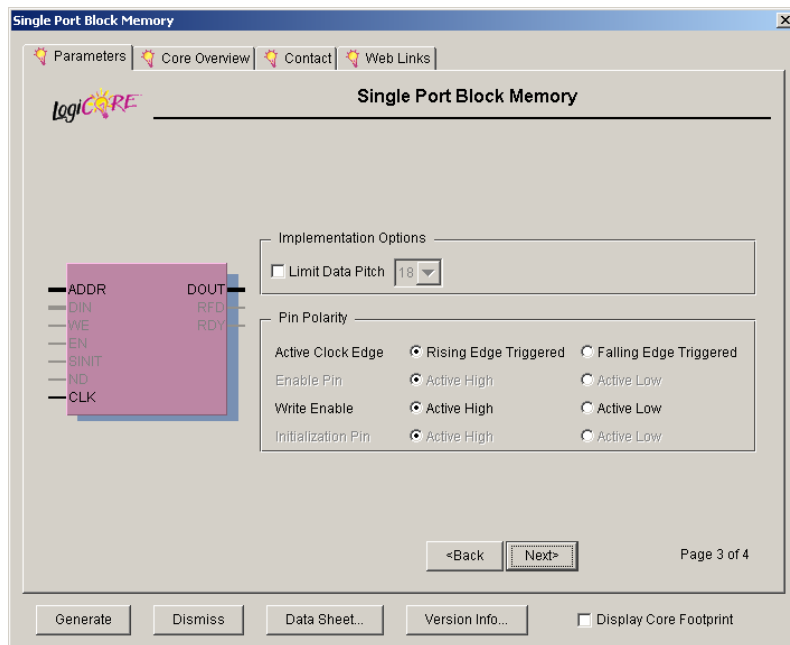
Memory size 里填好 ROM 的宽度和深度，上图中是一个 256*16BIT 的 ROM。

点击 NEXT 进入下图



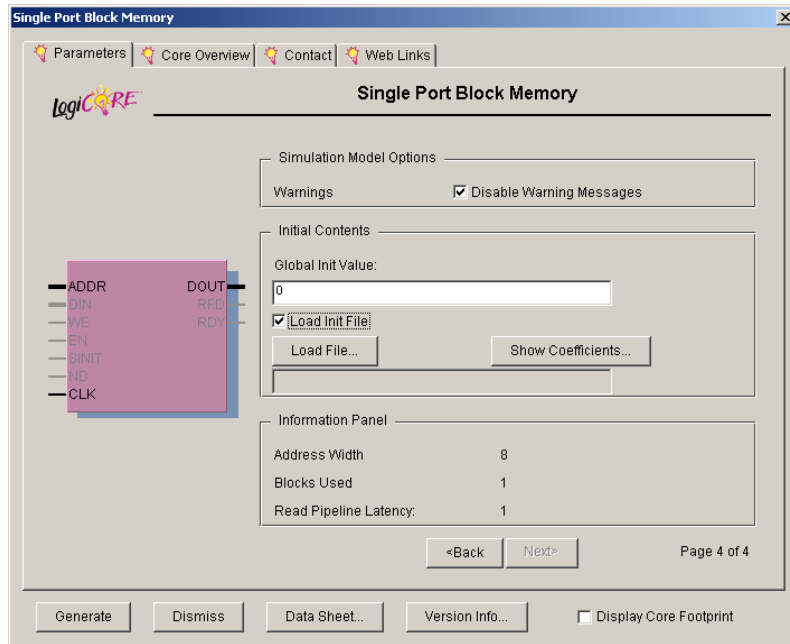
这里默认就可以了。

再 NEXT



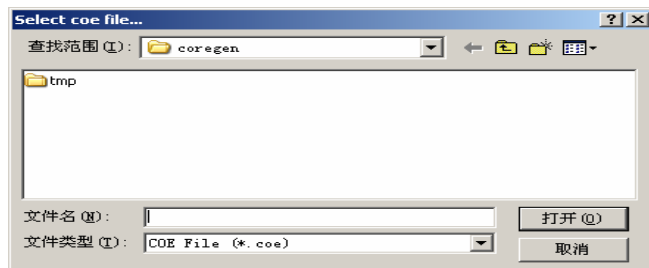
这里可以选择是 CLK 上升沿读数据还是下降沿读。

继续 NEXT



这里比较关键，要导入 ROM 的值

在 LOAD INIT FILE 项上打勾，点击 LOAD FILE...出现下图，要你选择一个文件。



这个文件就是你要放在 ROM 中的数据，文件的后缀名是.COE。你可以自己建一个空的文本文件，然后把后缀名该为.COE 即可。

文件内容的格式如下

```
MEMORY_INITIALIZATION_RADIX=10;
```

```
MEMORY_INITIALIZATION_VECTOR=
```

```
32767,
```

```
32766,
```

```
32766,
```

```
32765,
```

```
32764,
```

```
32763,
```

```
32761,
```

```
32759,
```

```
32757,
```

```
.....
```

```
32754;
```

注意：

第一行定义文件中的数据采用什么进制。可以使用 10 进制，2 进制，16 进制，数据大小不

能超过你定义的数据宽度

第二行照抄

第三行开始就是数据了，每个数据用逗号“，”隔开，可以不分行。建议分行写，便于统计个数。

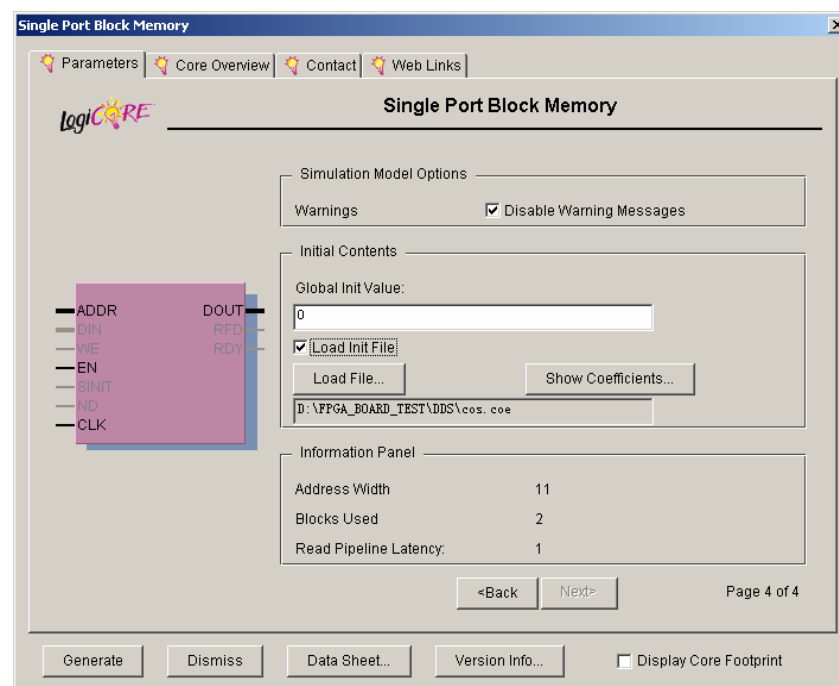
最后一个数据后用分号“；”结束。

数据的个数必须和你定义的数据深度相同，否则会出错。

写好这个文件后，保持。

然后在点击 LOAD FILE...，选择这个文件，点打开。

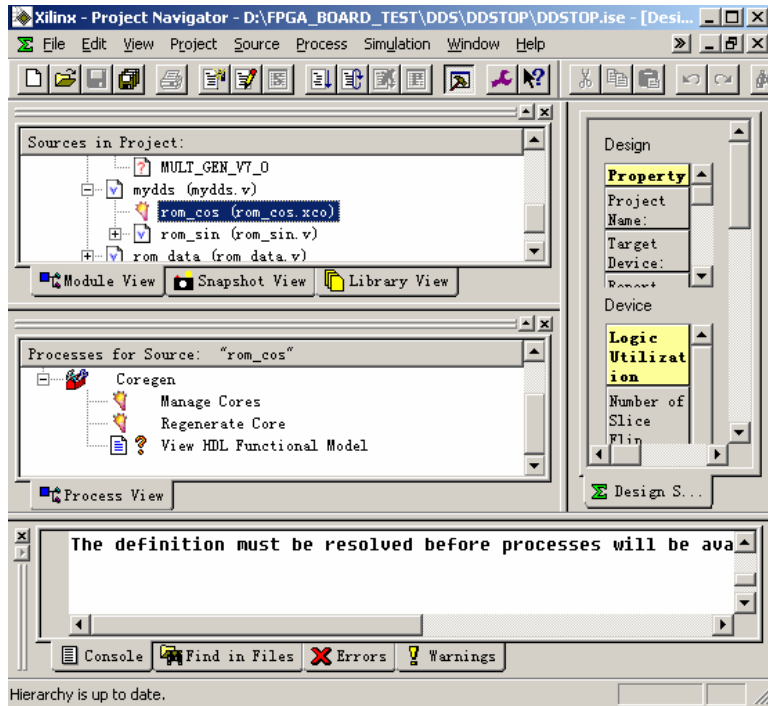
如果没有错误，可以点击旁边的 show coefficients 查看数据。如果数据很多 (>512)，建议不要使用此功能，可能会因数据太大而无法响应。



现在，可以点击 GENERATE，产生代码了。

产生的代码中有个.MIF 文件，就是 ROM 的数据的二进制文件。

其中的.V 文件或.VHD 就是生成的可综合的代码。

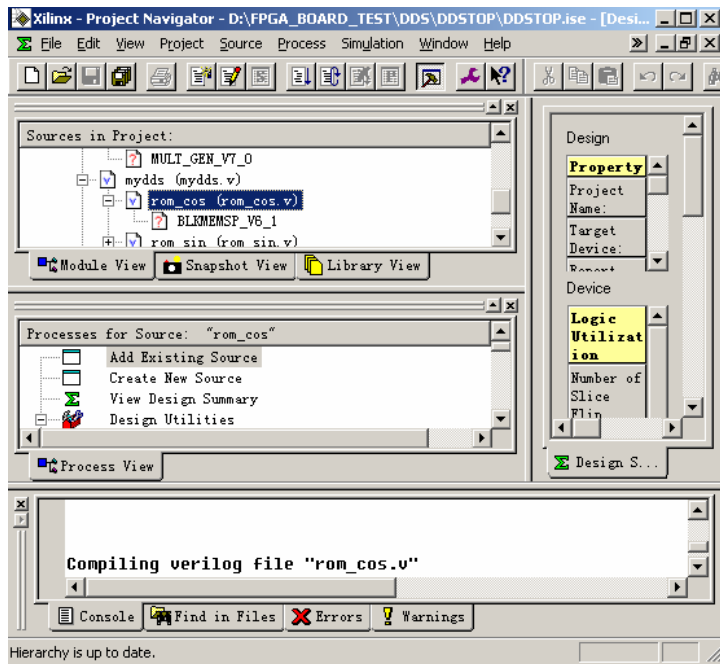


Hierarchy is up to date.

如上图的 ROM_COS.XCO, 选择它, 双击 VIEW HDL FUNCTIONAL MODEL 就可查看 verilog 代码。

接下来, 要删除 ROM_COS.XCO, 然后把 ROM_COS.V 添加进来。

接下来就可以综合了。



Hierarchy is up to date.