

ALTERA FPGA 设计之综合实验

liyh@Advanced Electronics System Lab. @201702

实验目的:

- 1、熟悉 ALTERA IP 核及 Signal TAP II;
- 2、了解 FPGA 基本结构中的存储器。
- 3、熟悉 VHDL test bench(测试平台/测试激励)的设计;
- 4、熟练掌握 Altera FPGA 的开发环境、设计步骤和流程。

实验内容:

一、实验步骤

按照“Altera FPGA 开发入门”中的完整流程:建立设计工程(注意工程名,顶层实体名及存放文件夹等工程参数的设置)、添加并编写源文件、综合与分析、仿真、及硬件验证等等步骤。

二、正弦信号发生器的设计

利用 Matlab 产生一个周期的非负正弦波,然后对其采样,取 1024 点 8 位数据。在 QuartusII 中利用 MegaWizard 生成一个 1024*8bits 的存储器 (ram/rom),并利用上述正弦波数据来初始化此存储器。编写 VHDL 程序,按一定的方式(如顺序)将存储器中的数据读出并输出到 DA 端口,编译,仿真。最后为工程添加一个 signaltap II 文件,设置参数,编译,无错后下载到目标板,并在 signaltap II 中观看输出到 DA 端口的数据及波形。

- 1、利用 matlab 产生一个周期正弦波并按.mif 格式写入文件里
 - 1) 打开 matlab
 - 2) 在 matlab 中,切换当前文件夹到 sin_table_matlab.m 所在的文件夹
 - 3) 打开 sin_table_matlab.m 并运行
 - 4) 最后会在当前的文件夹下产生存储器初始化文件 mystorage.mif。
 - 5) 也可以直接利用实验室提供的存储器初始化文件 mystorage.mif。

- 2、创建一个新 Quartus II 工程 (project)

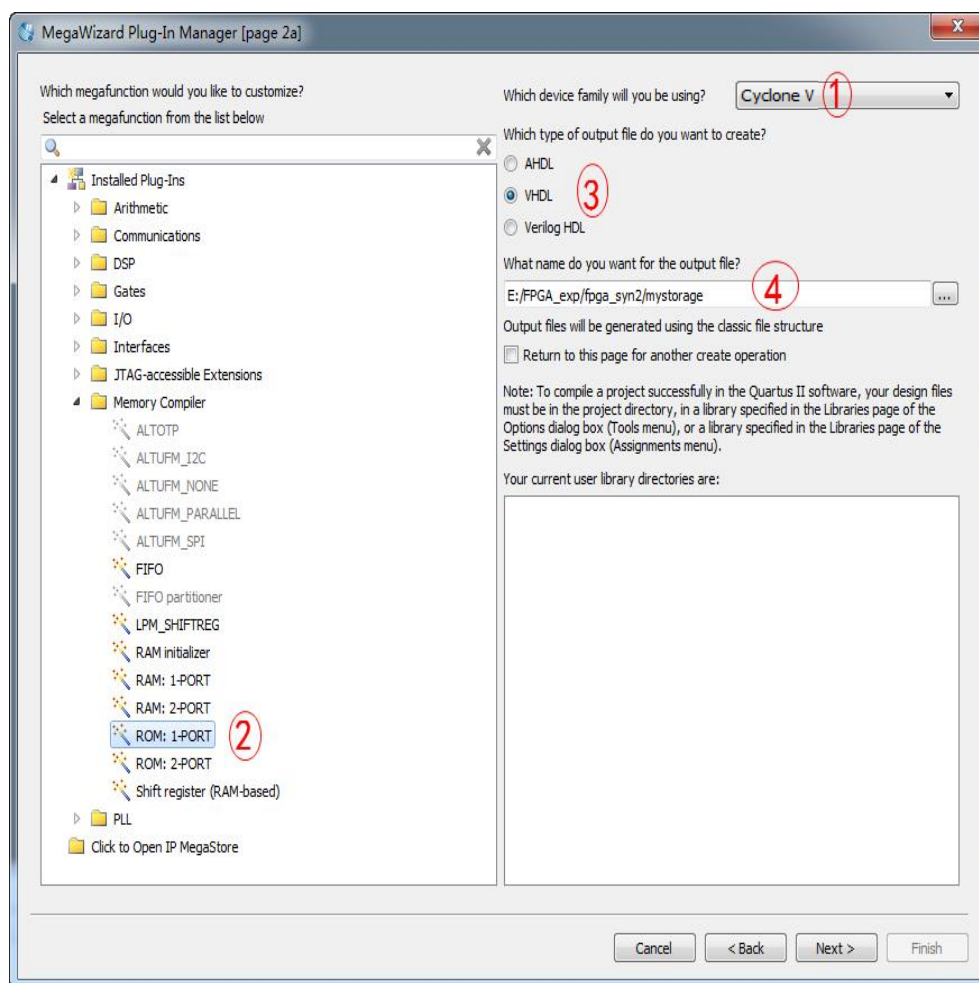
- 3、将存储器初始化文件(mystorage.mif)复制到 Quartus II 工程文件夹下。

- 4、利用 MegaWziard 生成 ROM 或 RAM 存储器

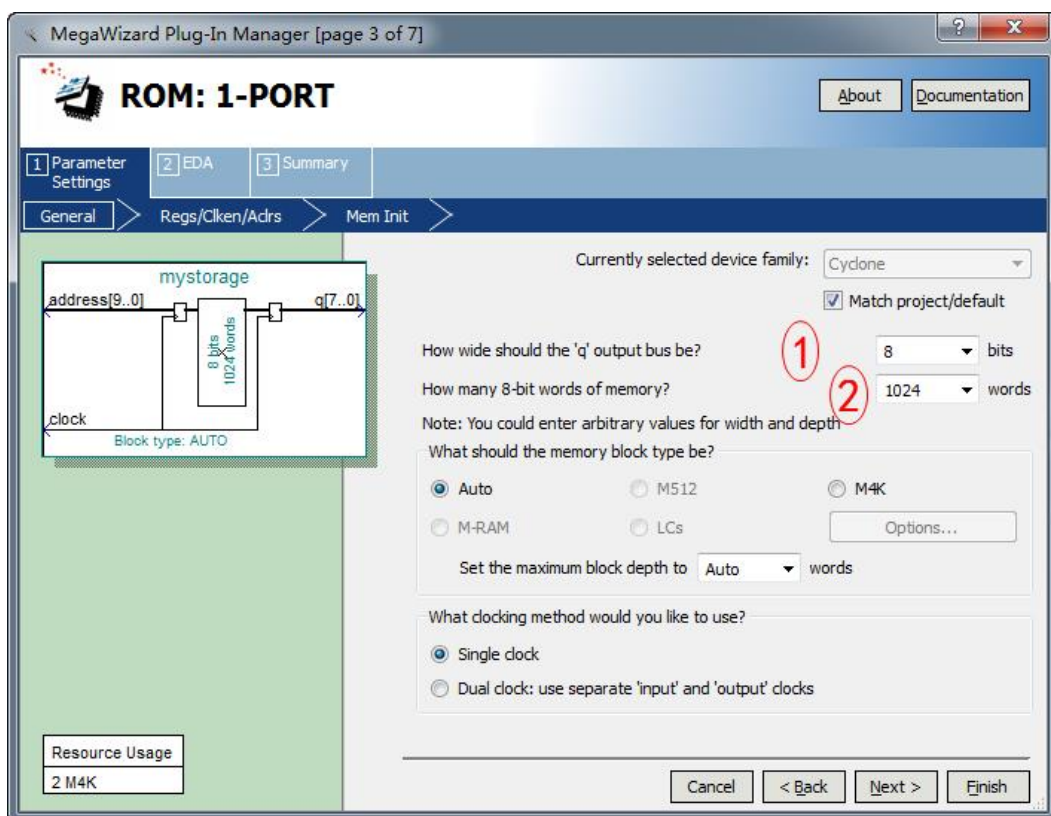
Altera 宏功能模块是复杂的高级构建模块,可在 Quartus II 设计文件中与逻辑门和触发器等基本单元一起使用。Altera 提供的参数化宏功能模块和 LPM 功能均为 Altera 器件结构做了优化。必须使用宏功能模块才可使用一些 Altera 专用器件的功能,如,存储器、DSP 块、PLL 等。有关 Altera 宏功能模块更详细的信息请参考 Quartus II 手册。下面给出本实验中所用

宏功能的基本步骤。

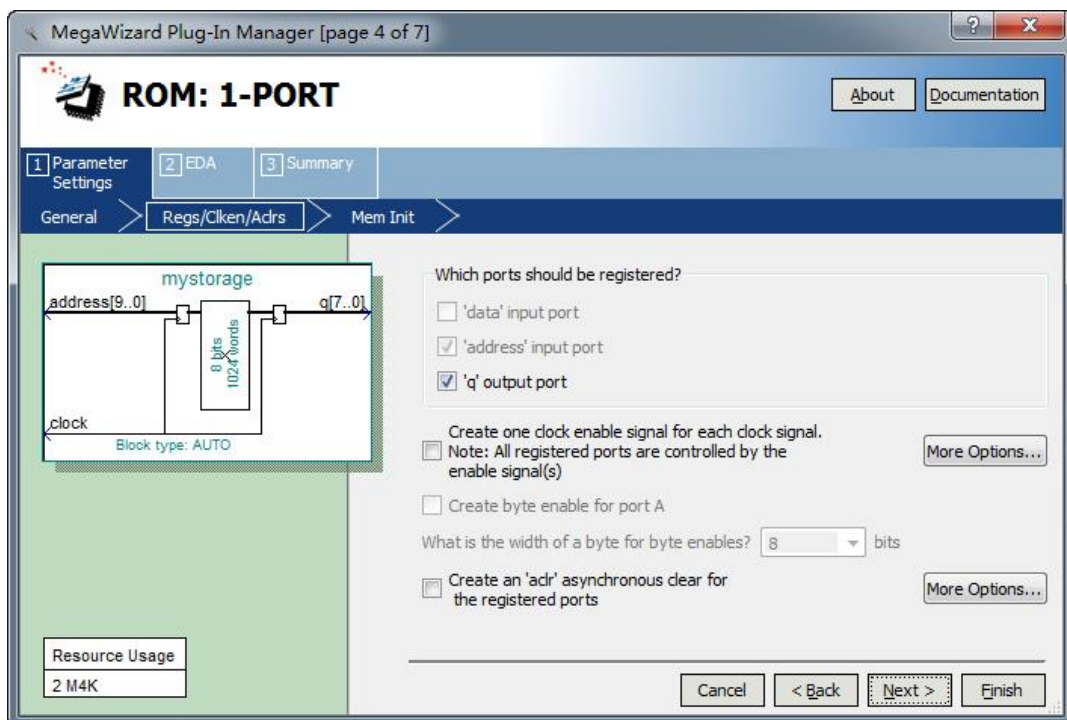
- 1) 启动 MegaWizard: 选择“Tools”菜单中的“MegaWizard Plug-In Manager”命令。
- 2) 在弹出的【page 1】窗口中选择“Which action do you want to perform”为“Create a new custom megafunction variation”。并单击“Next>”按钮进入下一步。
- 3) 在【page 2a】页面，要选择“Which device family will you be using”?为“Cyclone”；“Which megafunction would you like to customize?”为“Memory Compiler”中的“ROM: 1-PORT”；“Which type of output file do you want to create?”为“VHDL”；同时设置“What name do you want for the output file?”为“mystorage”到 Quartus II 的工程文件夹下，如下图所示。



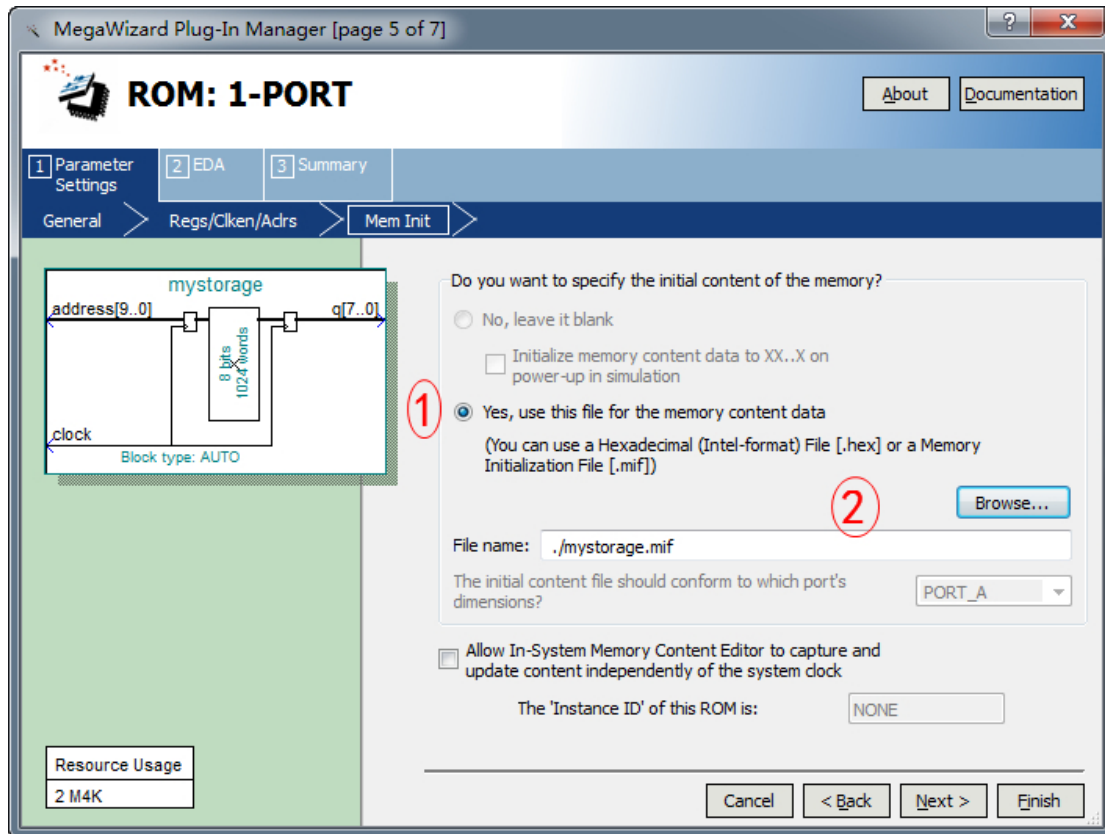
- 4) 在【page 3 of 7】页面，选择“How wide should the ‘q’ output bus be”为“8”；“How many 8-bit words of memory”为“1024”；其它默认设置。如下图所示。



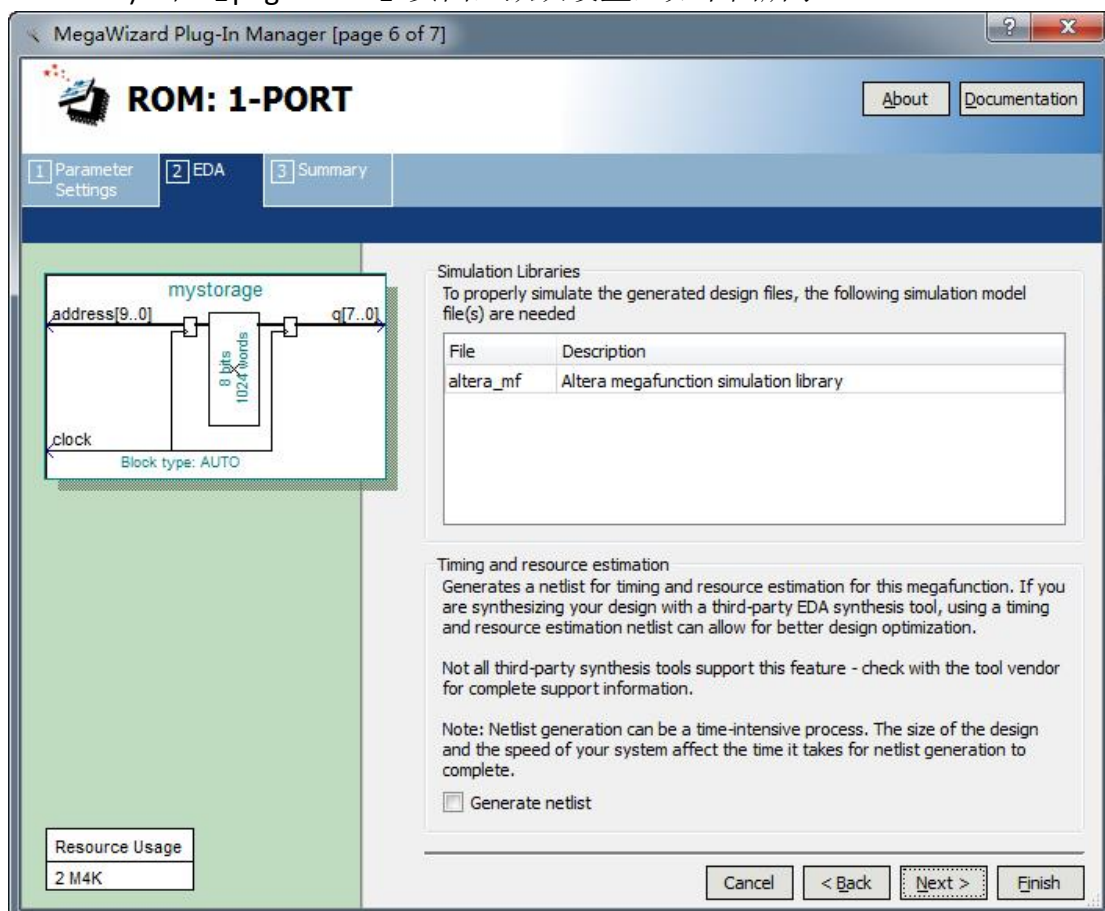
5) 在【page 4 of 7】页面，默认设置，如下图所示。



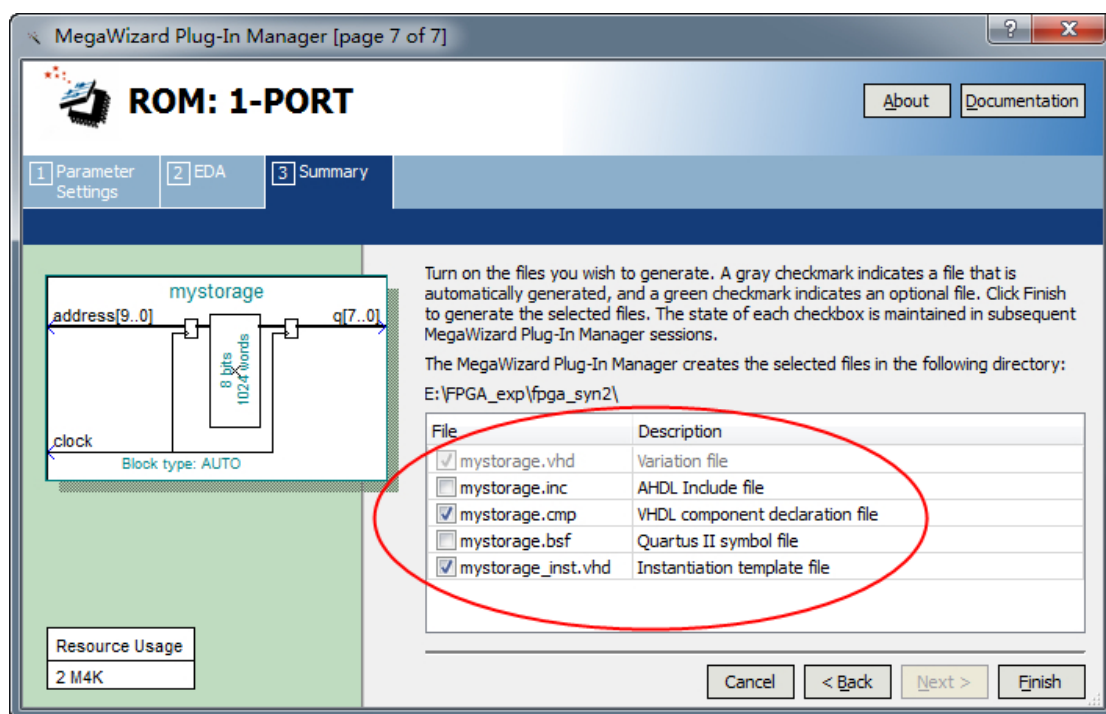
6) 在【page 5 of 7】页面，选择“Do you want to specify the initial content of the memory”为“Yes, use this file for the memory content data”，并通过“browse”按钮选择当前工程下的存储器初始化文件“mystorage.mif”，如下图所示。



7) 在【page 6 of 7】页面，默认设置，如下图所示。

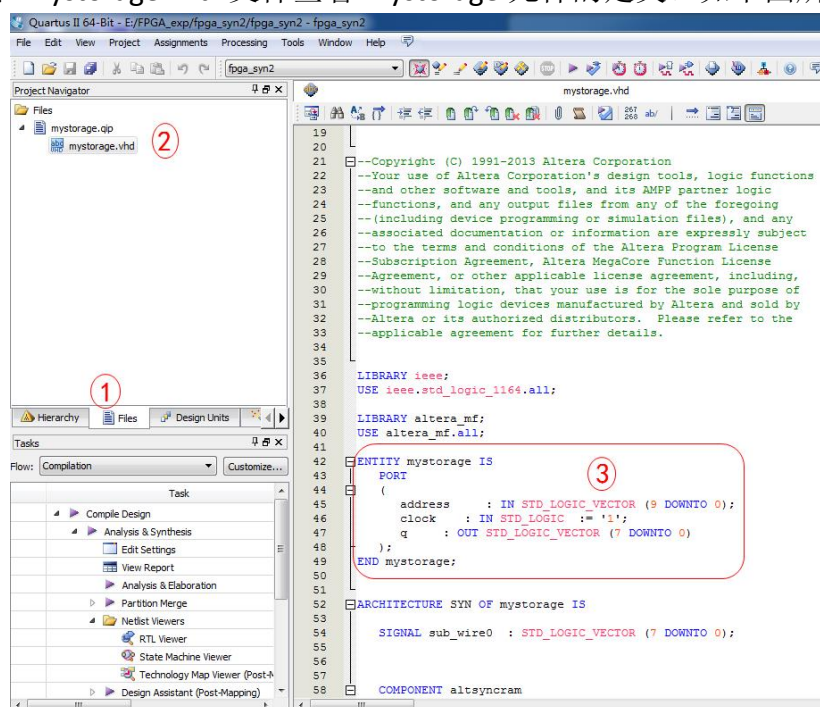


8) 在【page 7 of 7】页面，可以选择生成相关的文件，如下图所示。



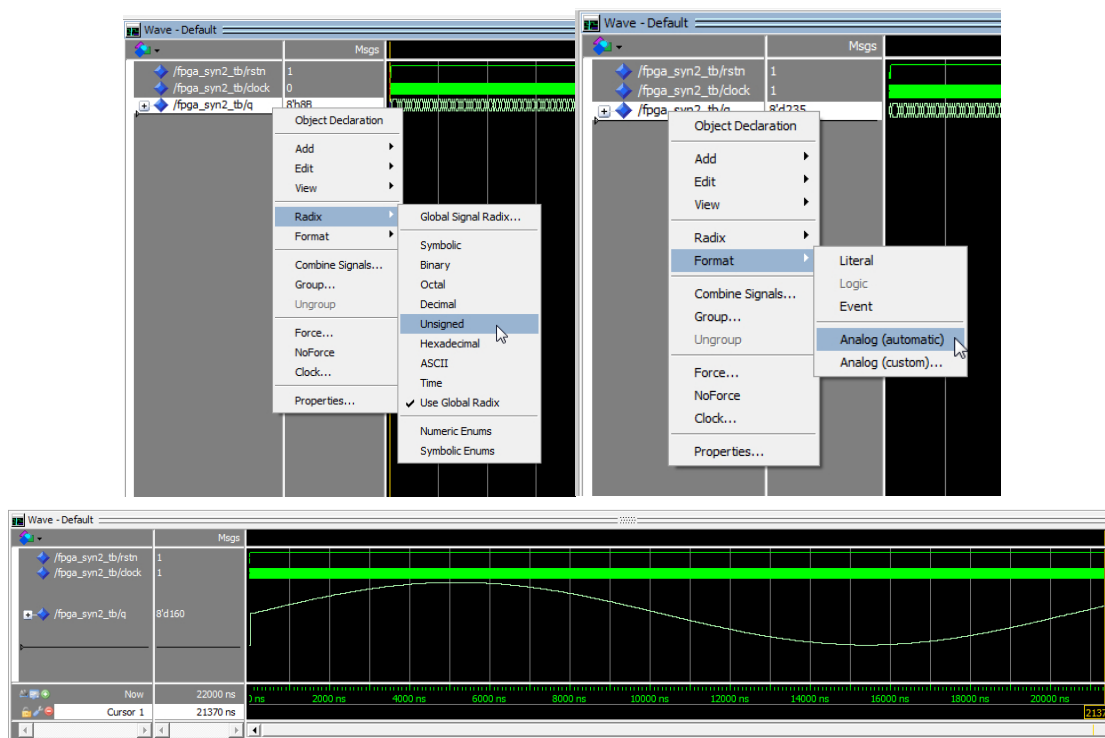
5、编写设计的顶层模块

- 1) 为工程添加顶层 vhdI 文件，使用 ieee 库中的 std_logic_1164 和 std_logic_unsigned 程序包。实体中的端口有复位和时钟输入，及 8 位的正弦输出。
- 2) 在顶层模块中声明并例化 mystorage 元件：在“Project Navigator”窗口选择“Files”页面，展开“mystorage.qip”后，双击“mystorage.vhd”，可以打开“mystorage.vhd”文件查看 mystorage 元件的定义。如下图所示。

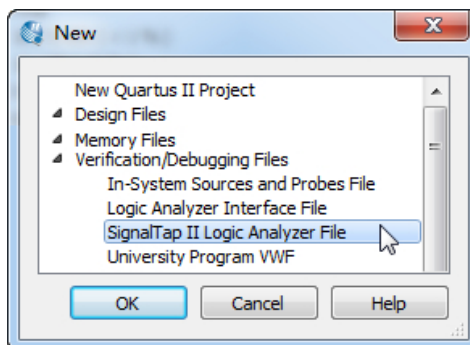


- 3) 再在顶层模块结构体中添加一个 process 来计数产生 mystorage 元件的地址，以便 mystorage 存储器中存储的正弦波数据依次的输出。
- 6、功能仿真：编写 vhdl test bench 文件，并设置相关参数后完成功能仿真。仿真时钟的周期用 20 ns，仿真的时间长度要大于 20 us。

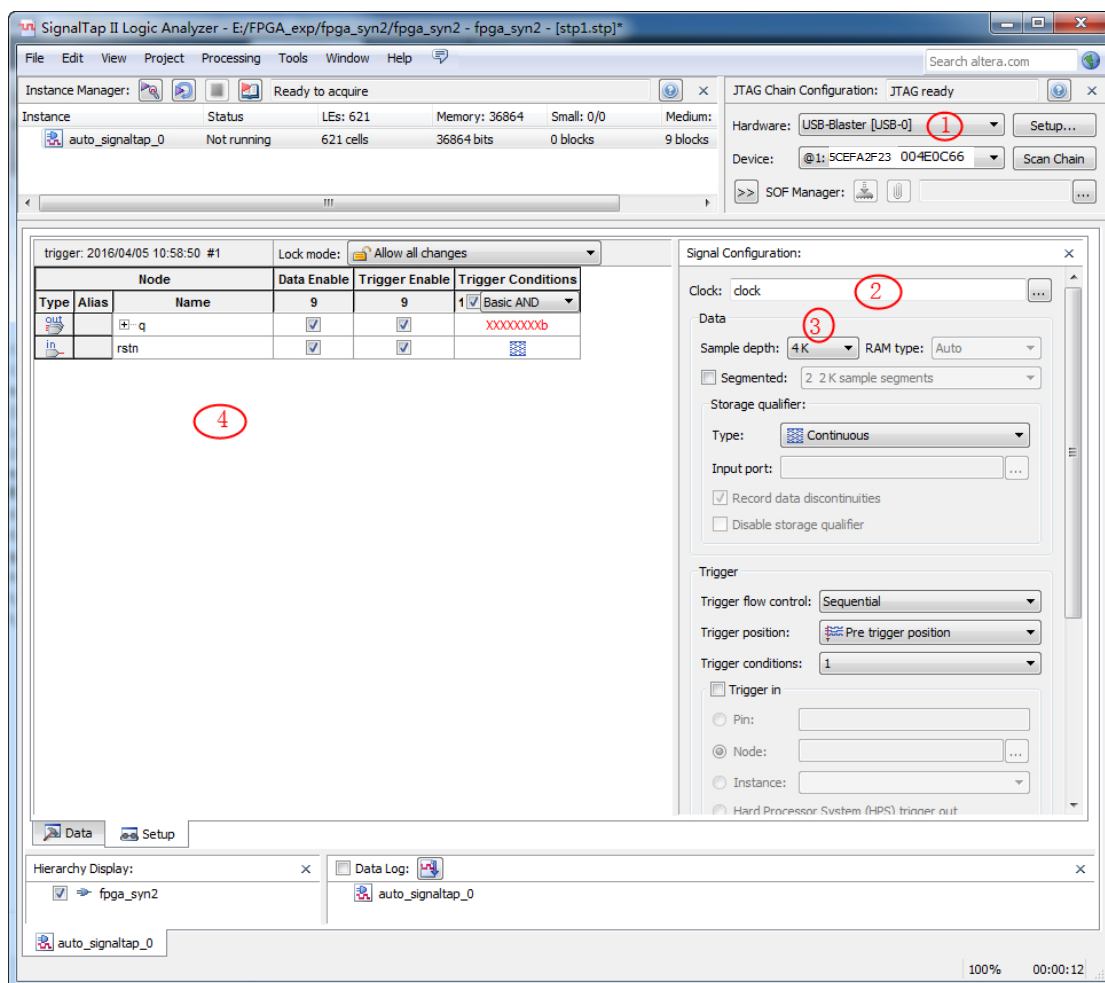
以模拟图的方式查看仿真的结果：①右键单击输出端口 q，选择 Radix 下的 Unsigned。②右键单击输出端口 q，选择 Format 下的 Analog(automatic)。



- 7、查看 RTL 级电路图
- 8、管脚分配：复位端口分配到输入数据开关 K 上；时钟端口分配到 50MHz 的晶振上；8 位的输出分配到 EMOD0 的 EMOD0[0~7]上。
- 9、为工程添加 Signal Tap II 模块
 - 1) 连接下载线，给 FPGA 目标板加电；
 - 2) 添加 Signal Tap II 模块：单击“File”菜单，选择“New...”命令，在打开的“New”窗口中选择“SignalTap II Logic Analyzer File”，进入 Signal Tap II Logic Analyzer 窗口。

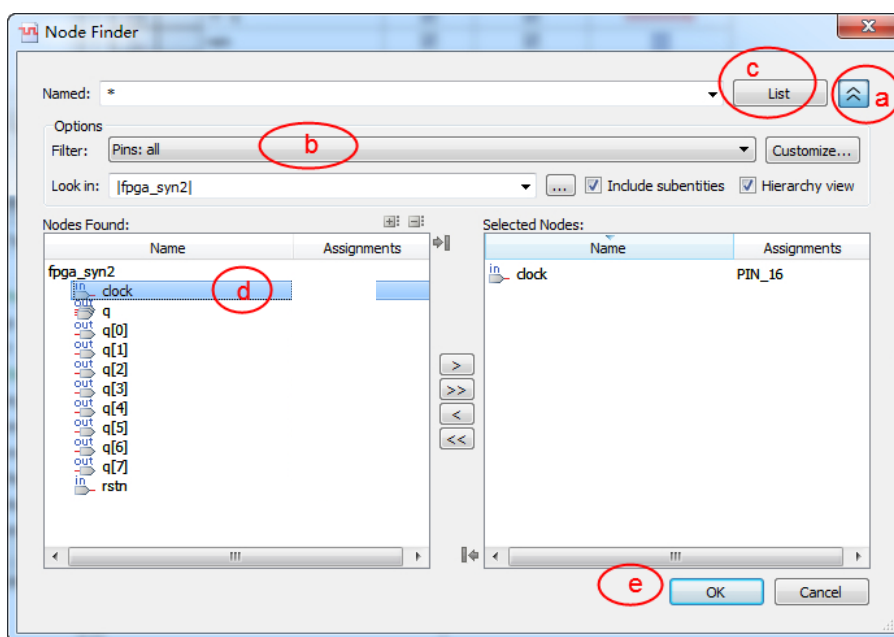


- 3) 设置 Signal Tap II



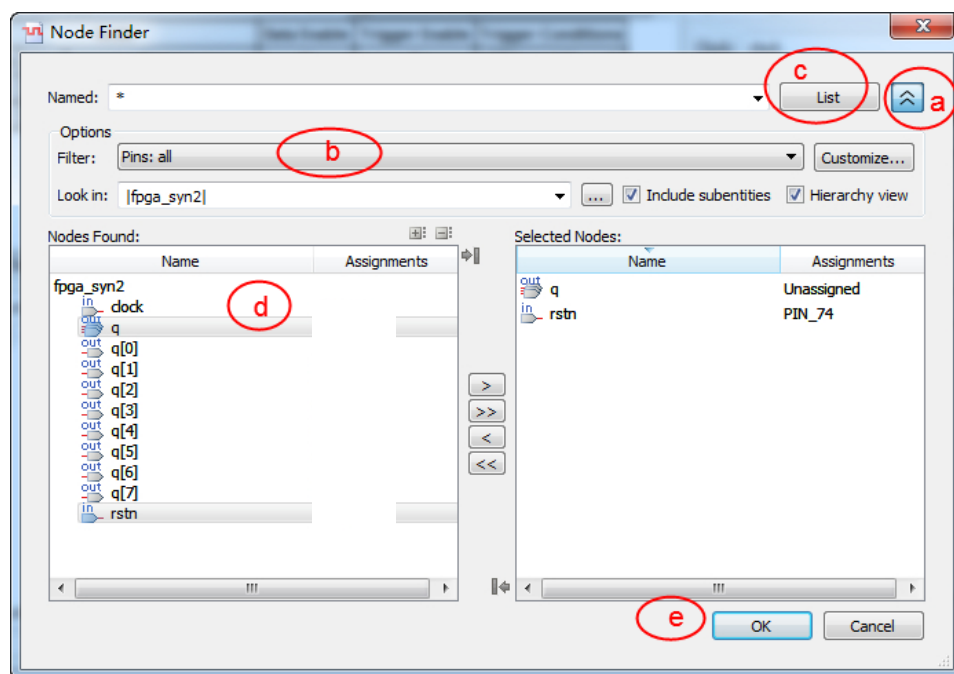
在上图的 Signal Tap II Logic Analyzer 窗口中，需要做如下设置：

- ① 如果“Hardware”和“Device”后的参数正确就不用设置了，否则要通过“Setup...”按钮完成设置。
- ② 通过“Clock:”最右侧的“...”按钮设置 Signal Tap II 的工作时钟为设计中的时钟。



在上图中先单击①处的按钮展开“Options”项，再在 Options 项中的 Filter 里选择“Pins:all”后，单击②处 List 按钮，然后在③处双击鼠标左键。最后单击“OK”按钮完成时钟的设置。

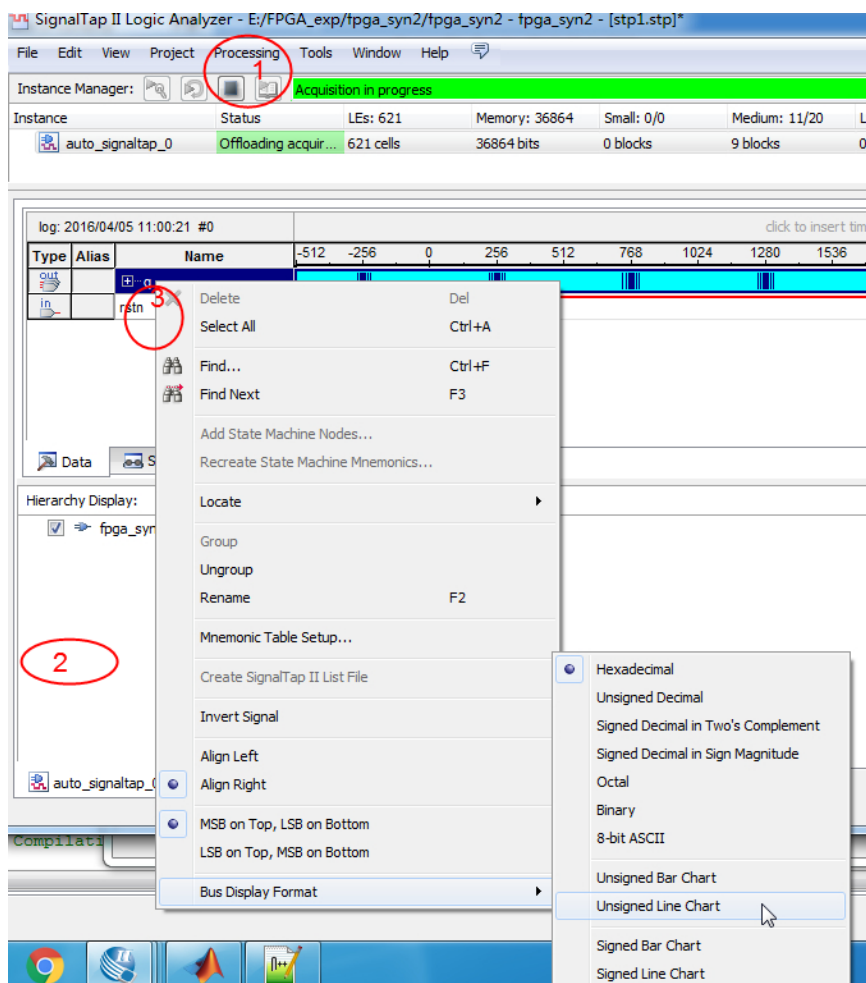
- ③ 设置 Sample Depth 为 4K，其它参数默认。
- ④ 在位置④处双击鼠标左键，在弹出的 Node Finder 窗口中，选择 q 和 rstn，方法同第②步时钟的设置。如下图所示。



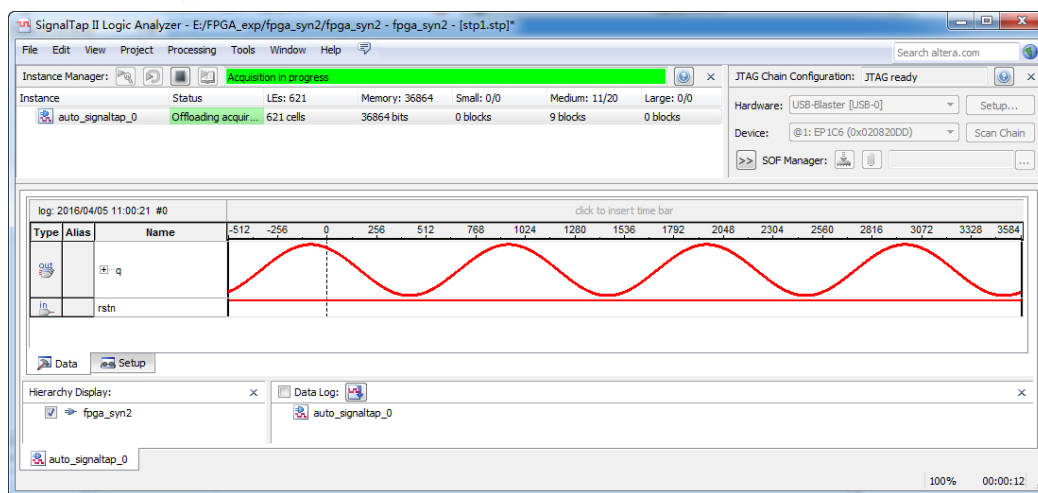
- ⑤ 最后通过“File”菜单中的“Save”或“Save As...”保存 SignalTap II 文件到当前工程文件夹下，并在弹出的“Do you want to enable SignalTap II File “文件名.stp” for the current project”窗口中，单击“Yes”按钮。

10、 通过 signalTap II 查看硬件验证结果

- 1) 重新编译（Start compilation）工程。
- 2) 下载设计到目标芯片（Programmer）。
- 3) 单击“Processing”菜单下的“Autorun Analysis”命令，启动 SignalTap II。如下图所示。
- 4) 然后将“setup”页面切换的“data”页面，如下图中的位置②。
- 5) 在下图中的位置③处的输出 q 上单击鼠标右键，选择弹出菜单中“Bus Display Format”子菜单下的“Unsigned Line Chart”命令。



6) 最后可以通过 SignalTap II 看到设计的硬件验证结果为连续的正弦信号输出。如下图所示。



三、 完成

如果你结束了今天的实验，请备份实验数据，关闭所有打开的软件后再关闭计算机；关闭实验盒的电源，并拆除所有相关的连线，恢复到实验之前的整齐样子，别留下任何物品。O(∩_∩)O 谢谢！