生成代码前准备工作

在从模型生成 HDL 代码前，你应该作如下工作：

- 在生成代码前，用 hdlsetup 功能（详见 Initializing Model Parameters with hdlsetup ）来对你要生成 HDL 代码的模型进行设置。
- 用 hdllib 功能来创建目前支持 HDL 代码生成的模块库（详见 Show Blocks Supported for HDL Code Generation）通过用这个库里的模块来构建模型，你的模型就会兼容HDL。
  支持的模块集合在今后的版本会改变，所以你每次安装新版本的本产品时应该重建支持的模块库。
- 用 **Run Compatibility Checker** 选项（详见 Selecting and Checking a Subsystem for HDL Compatibility）来检查你模型或者 DUT 的 HDL 兼容性并生成 HDL Code Check Report。
  你也可以调用 chekhdl 函数（见 checkhdl）来运行兼容性检查器。

练习简介

HDL Coder 支持生成代码，用户可以选择以下环境：

1. MATLAB 命令窗口支持用 makehdl，makehdltb 和其他函数生成代码
2. Simulink GUI (配置参数对话框和/或模型浏览器) 提供模型仿真参数/代码生成参数和函数的集成视图

这个实践练习通过在以上两种环境使用中相同的模型，介绍HDL 代码的生成和仿真机制。在这一系列步骤中，你将：

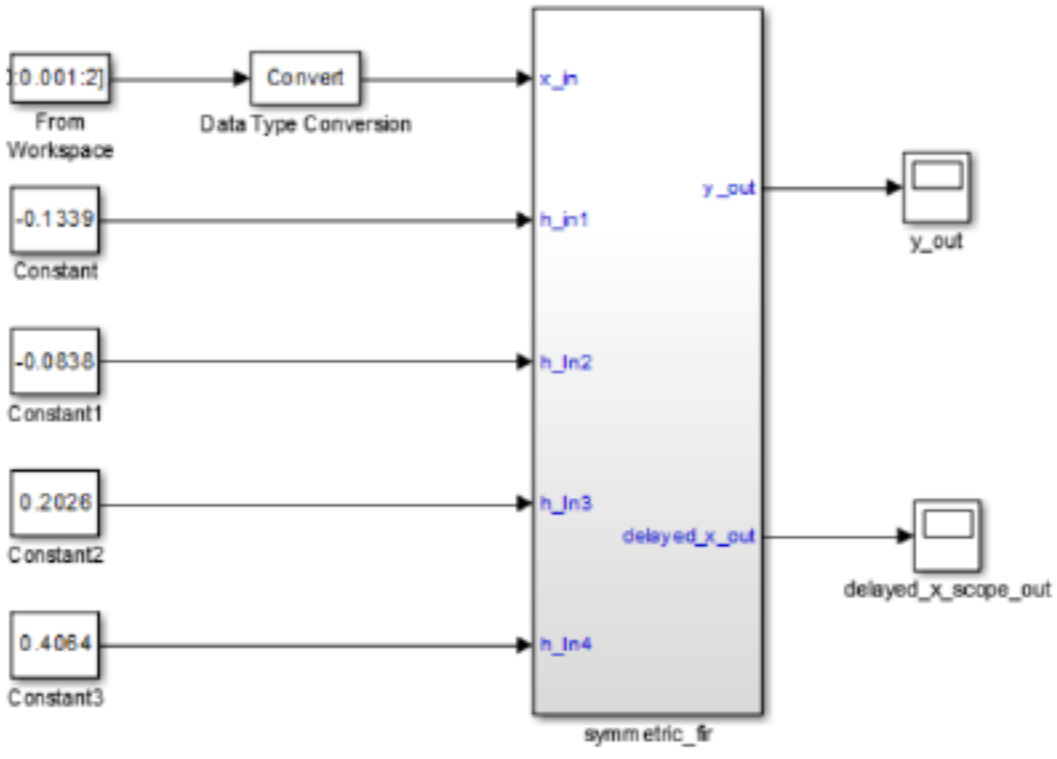
1. 构建一个简单模型用于生成代码
2. 从模型中的子系统生成代码
3. 生成适用于 Mentor Graphics ModelSim 仿真器的 VHDL 测试台来实现模型的仿真
4. 在仿真器中编译和执行模型和测试台
5. 用同样的模型生成并仿真 Verilog 代码
6. 检查模型和 HDL Coder 的兼容性

stir_fixed 模型

这些练习用 sfir_fixed 模型作为 HDL 代码生成源。该模型模拟一对称有限脉冲响应滤波器算法，通过定点数计算实现。

该模型里的模块支持 HDL 代码生成，并且模型参数已进行配置以适于代码生成。想了解更多关于模型准备以进行代码生成，参考 Prepare Simulink Model For HDL Code Generation.

下图为模型的顶层级别描述。



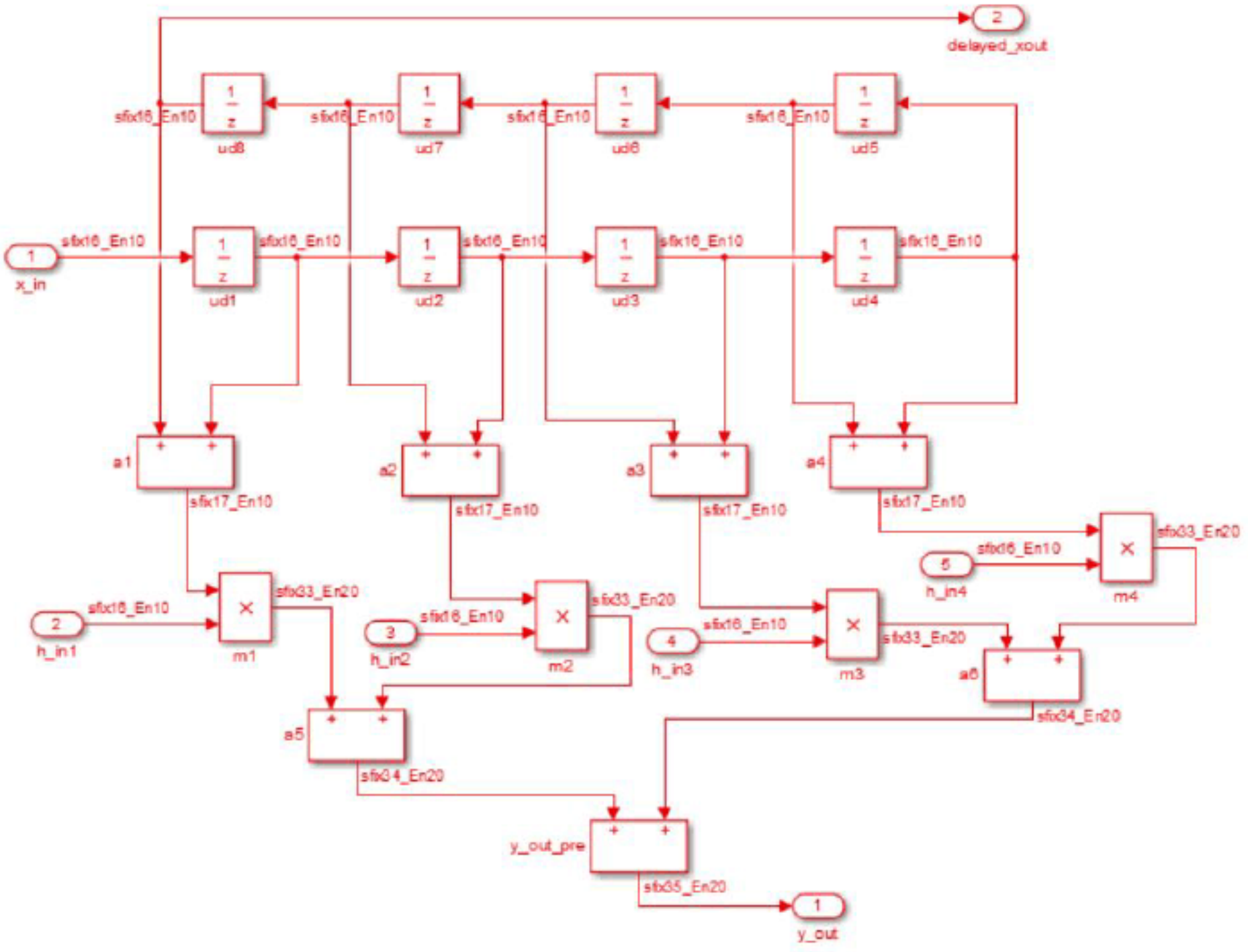
该模型采用了自

- 用于执行滤波 将从这个子系统被创建、测试，最终综合。
- 驱动该子系统的顶层模型元件是测试台。

顶层模型生成 symmetric_fir 子系统的 16 位定点输入信号。Signal From Workspace 模块生成一个用于滤波器的测试输入（激励）信号。四个常数模块提供滤波器系数。

Scope 模块仅用来仿真，这些虚拟的模块不会生成 HDL 代码。

下图描述了 symmetric_fir 子系统。



在接下来的例程里，你将会生成将 symmetric_fir 当成一个实体实现的 VHDL 代码。然后你会从顶层级别模型生成一个测试台。这个测试台用从Signal From Workspace 产生的激励数据驱动生成的实体完成指定的时钟步数。

用 HDL Workflow Advisor 生成代码（该法不能生成测试台代码 ）

这个例程展示了如何用 HDL Workflow Advisor 从 Simulink 模型生成 HDL 代码。

本例程中的模型，stir_fixed 已经被准备好用于代码生成。

本例程采用 Xilinx ISE 综合工具，我们假设你的工具路径已设置好。你也可以用Altera Quartus II 来进行本例程。

创建工作文件夹并复制模型

- 用 HDL Workflow Advisor 生成代码
- 执行 FPGA 综合和分析

创建工作文件夹并复制模型

1. 启动 MATLAB
2. 创建一名为 sl_hdlcoder_work 的文件夹。比如：

```
mkdir C:\work\sl_hdlcoder_work
```

sl_hdlcoder_work 文件夹将被用于保存例程中模型的拷贝和 HDL Coder 生成的文件夹及代码。这个文件夹的位置不重要，但是不能在 MATLAB 的文件夹树下。

3. 将 sl_hdlcoder_work 文件夹设置为工作文件夹。比如：

```
cd C:\work\sl_hdlcoder_work
```

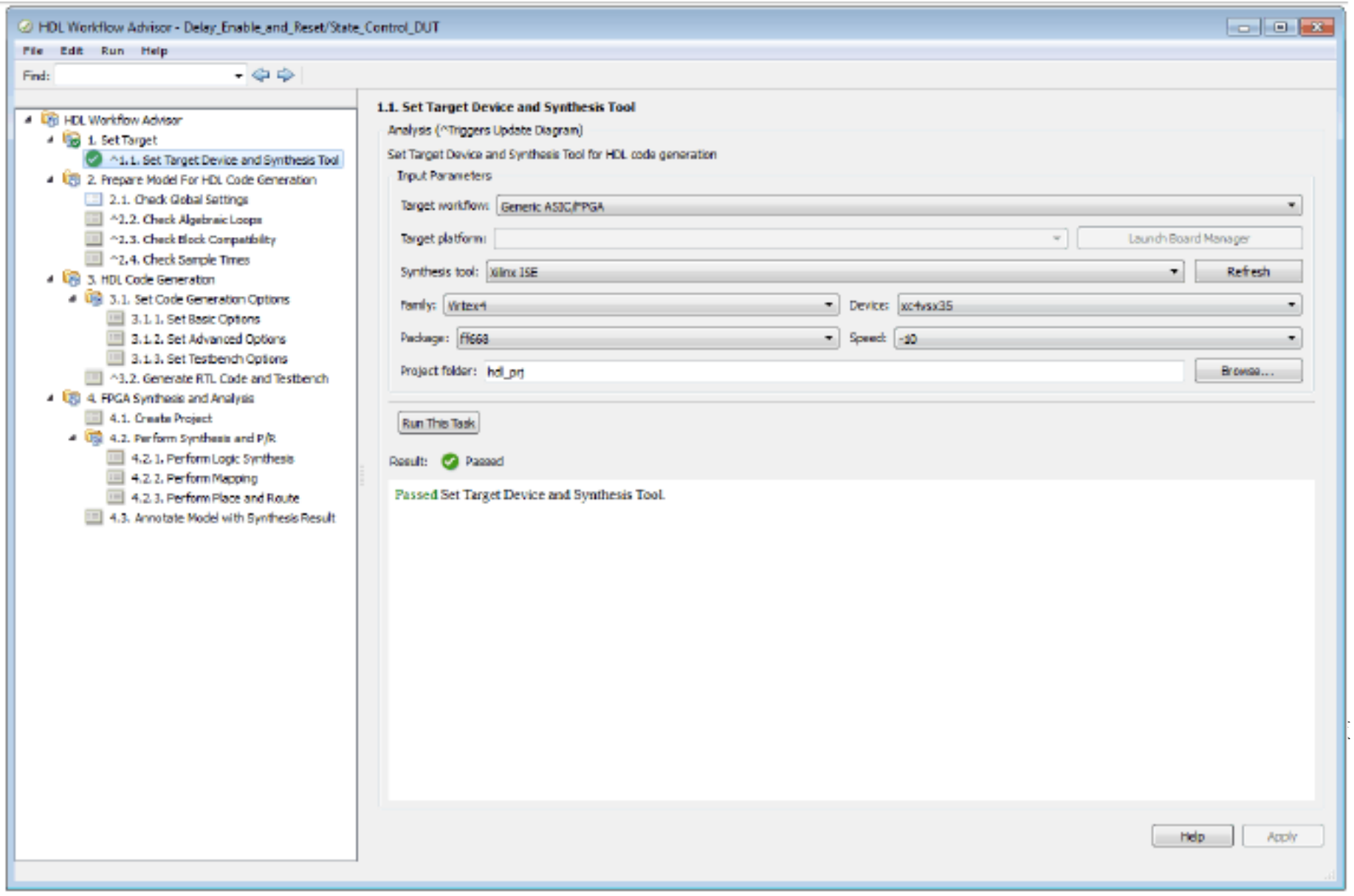
4. 打开 sfir_fixed 模型。

```
sfir_fixed
```

5. 将 sfir_fixed 复制到 sl_hdlcoder_work 文件夹下。

用 HDL Workflow Advisor 生成代码

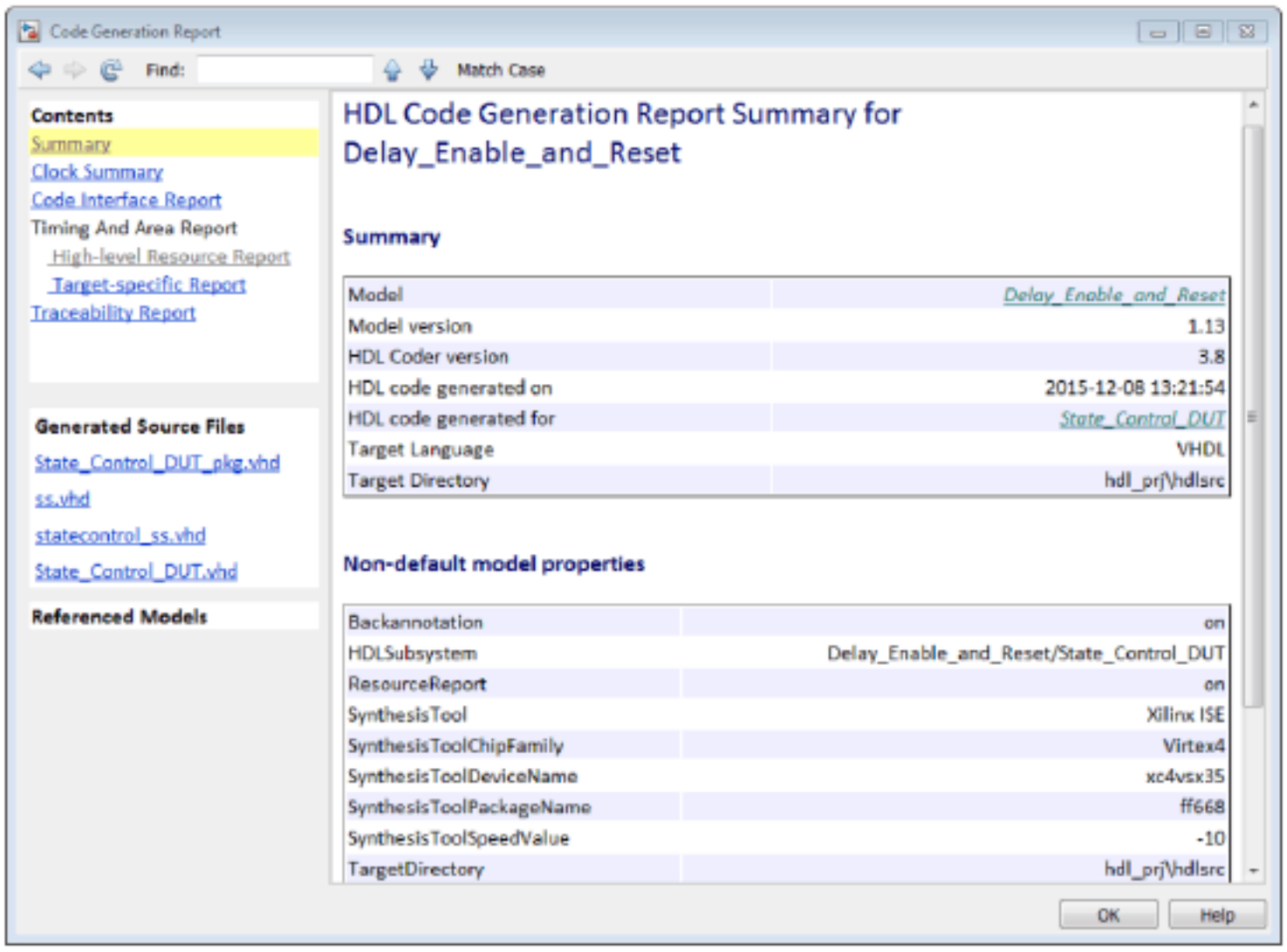
1. 右键点击 symmetric_fir 子系统，选择 **HDL Code > HDL Workflow Advisor**。
2. 在步骤 **Set Target > Set Target Device and Synthesis Tool**, 在 **Synthesis tool** 选择 **Xilinx ISE** 并点击 **Run This Task**。



- 在 **Language**, 选择 **Verilog**.
- 使能 **Generate traceability report**.
- 使能 **Generate resource utilization report**.

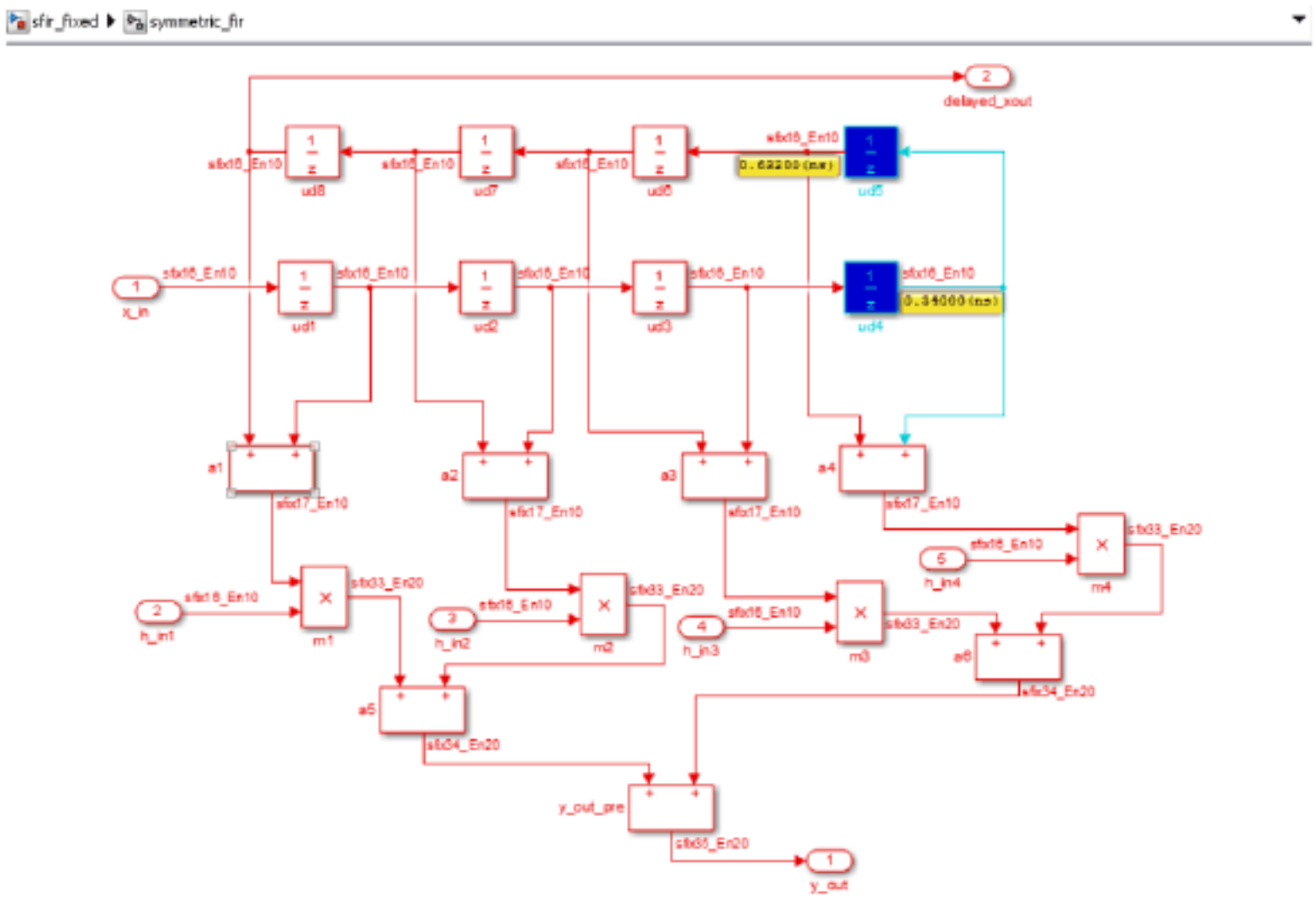
3. 浏览 **Optimization** 和 **Coding style** 标签下的可用选项。这些选项可以用来修改所生成代码的实现和格式。
4. 右键点击 **HDL Code Generation > Generate RTL Code and Testbench step**， 选择 **Run to Selected Task**。

   代码生成报告自动被打开，它包括资源使用和可追溯性报告。资源使用报告描述你的设计对硬件资源的使用情况。可追溯性报告允许你在模型和生成的代码之间操作。



执行 **FPGA**

1. 在 **…ute** 任务下
2. 右键点击 **Annotate Model with Synthesis Result** 并选择 **Run to Selected Task**。
3. 观察模型中被注释的关键步骤。关键步骤以青色显示。



用命令行生成代码

- 概览
- 创建一文件夹和本地模型文件
- 用 hdlsetup 初始化模型参数
- 从子系统生成 VHDL 实体（VHDL Entity）
- 生成 VHDL 测试台代码（Test Bench Code）
- 检验生成的代码
- 生成一个 Verilog Module 和测试台

概览

这一例程提供代码和测试台生成指令、其参数和代码生成器创建的文件的详细步骤介绍。该例程假设读者已经熟悉了例程模型(参见 The sfir_fixed Model)。

创建一文件夹和本地模型文件

创建案例模型的一本地拷贝并将其保存在工作文件夹下，如下所示：

1. 启动 MATLAB。
2. 创建一名为 sl_hdlcoder_work 的文件夹。比如：

```
mkdir C:\work\sl_hdlcoder_work
```

sl_hdlcoder_work 文件夹将被用于保存例程中模型的拷贝和 HDL Coder 生成的文件夹及代码。这个文件夹的位置不重要，但是不能在 MATLAB 的文件夹树下。

3. 将 sl_hdlcoder_work 文件夹设置为工作文件夹。比如：

```
cd C:\work\sl_hdlcoder_work
```

4. 打开例程模型，在 MATLAB 命令行下输入如下指令：

```
sfir_fixed
```

5. 在 Simulink 中，选择 As，将 sfir_fixed 模型另存一本地拷贝到工作文件夹中。
6. 保持 sfir_fixed 打开，继续下面的进程。

用 **hdlsetup** 初始化模型参数

在生成代码前，模型必须进行配置。你可以使用 hdlsetup 指令从而避免手动配置模型。Hdlsetup 指令使用 set_param 函数来配置模型以快速并可靠地生成 HDL 代码。

配置模型以进行代码生成：

1. 在 MATLAB 命令行，输入:

```
hdlsetup('sfir_fixed')
```

2. 在 **File** 菜单下选择 **Save**，从而保存模型及其新的设置。

在继续进行代码生成前，考虑 hdlsetup 在模型中应用的设置。

Hdlsetup 配置 HDL Coder 推荐或要求的求解器（Solver）选项，它们是:

- **类型（Type）**: 固定步长（Fixed-step）。 (HDL Coder 目前支持特定条件下的可变步长求解器。参见 hdlsetup)
- **求解器（Solver）**: 离散（Discrete，无连续状态）。也可以选择其他的固定步长求解器，但用于仿真离散系统这一选项通常是最佳的。
- **任务模式（Tasking mode）**: 单任务（SingleTasking）。 HDL Coder 目前不支持执行多任务的模型。

不要将 **Tasking mode** 设置为 Auto。

Hdlsetup 也配置模型开始和结束时间以及固定步长，如下所示:

- **开始时间（Start Time）**: 0.0 s
- **结束时间（Stop Time）**: 10 s
- **固定步长（Fixed step size）** (基本采样周期时间): auto

如果 **Fixed step size** 被设置为自动（auto），步长大小就根据模型中指定的采样时间被自动选择 。在示例模型中，只有 Signal From Workspace 模块指定了明确的采样时间（1s）；其余模块继承这一采样时间。

模型的开始和结束时间决定了测试台仿真总时长。这也进一步决定了产生的用于为生成的测试台提供激励和输出数据的数据阵列的大小。对于例程模型，10 秒的测试数据计算不会花费很多时间。更加复杂的模型的采样值的计算可能很费时间。在这种情况下，就需要考虑减少仿真时间。

经 hdlsetup 配置的其余参数控制错误严重性级别，数据纪录和模型显示选项。如果想查阅与 hdlsetup 有关的完整模型参数集合，在 MATLAB Editor 中打开 hdlsetup.m 。

由 hdlsetup 提供的模型参数默认值是有用的，但对用户的模型来说并不一定是最优的。比如，hdlsetup 设置的 **Simulation stop time** 默认值为 10s。对例程模型 sfir_fixed 的测试 来说， 1000s 的仿真时间会更加现实。如果想修改仿真时间，在 Simulink 窗口中 **Simulation stop time** 区域输入目标值。

模型参数概要请参照 Smulink 资料中 Model and Block Parameters 区域的 Model Parameters 表格。

从子系统生成 VHDL 实体（VHDL Entity）

在这一部分，你将使用 makehdl 函数从例程模型的 symmetric_fir 子系统生成 VHDL 实体的代码。makehdl 也生成用于第三方 HDL 仿真和综合工具的脚本文件。

makehdl 允许你指定多种特征用以控制生成的代码的各种特性。在这一例程中，你将使用 makehdl 默认的特征参数。

在生成代码前，请确定已完成了创建一文件夹和本地模型文件及用hdlsetup 初始化模型参数的步骤。

生成代码：

1. 在 MATLAB 窗口下的桌面菜单选择 Current Folder。这样会显示 Matlab 的 Current Folder browser，使你能快速打开工作文件夹和将要在其中生成的文件。
2. 在 Matlab 命令行，输入指令：

   makehdl('sfir_fixed/symmetric_fir')

   这条指令命令 HDL Coder 从 sfir_fixed 模型下的 symmetric_fir 子系统用默认参数生成代码。
3. 在代码生成时，HDL Coder 显示进程消息。这个进程应该以下面的信息结束：

   ### HDL Code Generation Complete.

   观察到进程消息中生成的文件的名字被超链接。在代码生成完成时，你可以点击这些超链接来查看 MATLAB Editor 里的文件。

   Makehdl 在代码生成前编译模型。代码生成后模型的外观可能发生改变，这取决于模型的显示选项（比如端口数据类型等）。
4. Makehdl 默认生成 VHDL 代码。代码文件和脚本被写入目标文件夹。默认的目标文件夹是工作文件夹下的子文件夹，名称为 hdlsrc。

   在 Current Folder browser 中可以看到一个 hdlsrc 文件的文件夹图标。要看生成的代码和脚本文件，双击这个 hdlsrc 文件夹图标。
5. Makehdl 在 hdlsrc 文件夹下生成的文件有：

- symmetric_fir.vhd：VHDL 代码。该文件包含一个 entity 定义和一个用于执行 symmetric_fir 滤波器的 RTL 架构。
- symmetric_fir_compile.do：Mentor Graphics ModelSim 编译脚本（vcom command）用于编译生成的 VHDL 代码。
- symmetric_fir_synplify.tcl：Synplify 综合脚本。
- symmetric_fir_map.txt：映射文件。这个报告文件将生成的实体（或模块）映射到用于生成它们的子系统（参照 Trace Code Using the Mapping File）。

6. 要在 Matlab Editor 中查看生成的 VHDL 代码，在 Current Folder browser 中双击 symmetric_fir.vhd 文件夹图标。
7. 在进行下一步之前，关闭在编辑器中打开的文件。然后，点击Current Folder browser 中的 Go Up One Level 按钮来将当前文件夹重新设置为 sl_hdlcoder_work 文件夹。
8. 保持 sfir_fixed model 处于打开状态，进行下面的步骤。

生成 VHDL 测试台代码

在这一部分，你使用测试台生成函数 makehdltb 来生成 VHDL 测试台代码。这个测试台的目的是驱动和验证上一部分生成的 symmetric_fir 实体的运行。生成的测试台包括：

- 由信号源产生的连接到被测 entity 源激励数据。