

HPS 仿真支持

该部分介绍硬核处理器系统 (HPS) 组件的仿真支持。HPS 仿真模型支持 HPS 和 FPGA 内核逻辑之间的接口，包括：

- HPS 和 FPGA 内核逻辑之间的大部分接口的总线功能模型 (BFMs)
- HPS SDRAM 存储器的仿真模型

HPS 仿真支持不包含 HPS 中实现的模块，例如 ARM® Cortex™-A9 MPCore 处理器。

当在 Qsys 系统集成工具中例化 HPS 组件时，您需要指定仿真支持文件。当使能一个特定的 HPS-FPGA 接口时，Qsys 在生成过程期间提供相应的模型。请参考第 29 - 8 页的“[仿真流程](#)”以便了解关于仿真流程的详细说明。



要了解关于例化组件的更多信息，请参考 *Cyclone® V 器件手册* 第 3 卷的 *Instantiating the HPS Component* 章节。

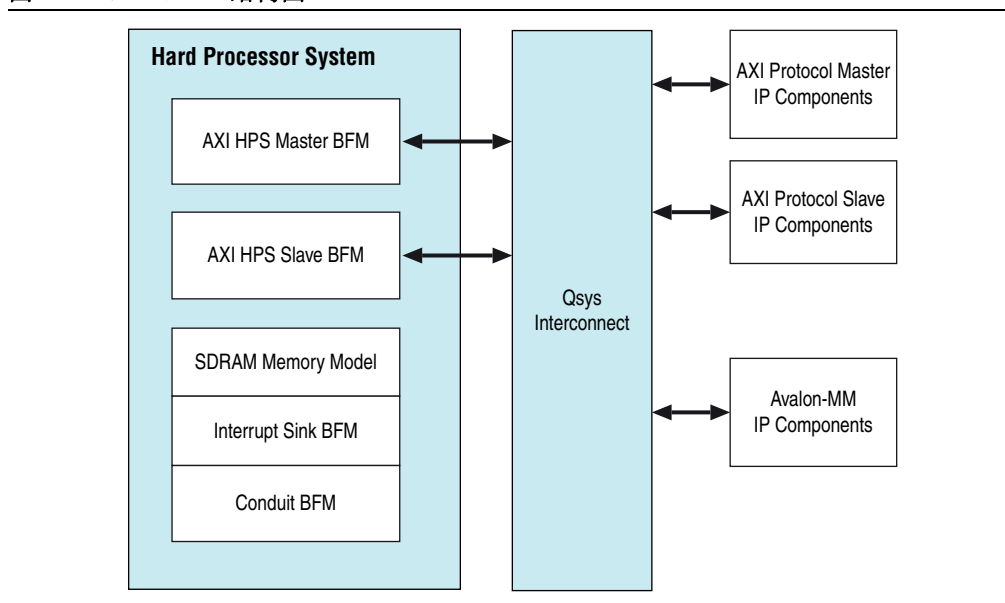
HPS 仿真支持使您能够开发和验证与 HPS 组件相连接的您自己的 FPGA 软逻辑或知识产权 (IP)。

仿真模型支持以下接口：

- 时钟和复位接口
- FPGA-to-HPS 高级微处理器总线体系结构 (AMBA®) 高级可扩展接口 (AXI™) 从接口
- HPS-to-FPGA AXI 主接口
- 轻型 HPS-to-FPGA AXI 主接口
- FPGA-to-HPS SDRAM 接口
- 微处理器单元 (MPU) 通用 I/O 接口
- MPU 备用和事件接口
- 中断接口
- 直接存储器访问 (DMA) 控制器外设请求接口
- 调试高级外设总线 (APB™) 接口
- 系统跟踪宏单元 (STM) 硬件事件
- FPGA 交叉触发接口
- FPGA 跟踪端口接口

第 29 - 2 页的图 29 - 1 显示了 HPS 中的 BFM。

图 29 - 1. HPS BFM 结构图



HPS BFM 使用从 Altera® BFM 应用编程接口 (API) 中调用的标准函数，在该部分以下的内容中有详细介绍。

要了解关于 BFM API 的更多信息，请参考 *Avalon Verification IP Suite User Guide* and the *Mentor Verification IP Altera Edition User Guide*。

HPS 仿真仅支持 Verilog HDL 或 SystemVerilog 仿真环境。

时钟和复位接口

要了解关于时钟和复位接口的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *HPS 组件接口* 章节中的“存储器映射接口”。

时钟接口

Qsys 对 HPS 组件的每个时钟输出接口生成时钟源 BFM。对于 HPS-to-FPGA 用户时钟，当在 Qsys 中例化 HPS 组件时，在 **HPS Clocks** 页面的 **User clock frequency field** 中指定 BFM 时钟速率。

HPS-to-FPGA 跟踪端口接口单元对 FPGA 生成一个时钟输出，名称为 h2f_tpiu_clock。在仿真中，时钟源 BFM 也代表该时钟输出的行为。

表 29 - 1 列出了包含 BFM 实例名称的所有 HPS 时钟输出接口。

表 29 - 1. HPS 时钟输出接口仿真模型

接口名称	BFM 实例名称
h2f_user0_clock	h2f_user0_clock
h2f_user1_clock	h2f_user1_clock
h2f_user2_clock	h2f_user2_clock
h2f_tpiu_clock	h2f_tpiu_clock

Altera 时钟源 BFM 应用编程接口 (API) 可应用于表 29 - 1 中列出的所有 BFM。您的 Verilog 接口使用相同的 API，可以在不同的实例名称传递。

Qsys 不对 FPGA-to-HPS 时钟输入接口生成 BFM。

复位接口

HPS 复位请求和握手接口被连接到 Altera conduit BFM 来进行仿真。表 29 - 2 列出了每个接口的名称。您可以使用表 29 - 2 中列出的 API 监控复位请求接口状态更改或设置接口。

表 29 - 2. HPS 复位输入接口仿真模型

接口名称	BFM 实例名称	API 功能名称
f2h_cold_reset_req	f2h_cold_reset_req	get_f2h_cold_rst_req_n()
f2h_debug_reset_req	f2h_debug_reset_req	get_f2h_dbg_rst_req_n()
f2h_warm_reset_req	f2h_warm_reset_req	get_f2h_warm_rst_req_n()
h2f_warm_reset_handshake	h2f_warm_reset_handshake	set_h2f_pending_rst_req_n() get_f2h_pending_rst_ack_n()

表 29 - 3 列出了包含 BFM 实例名称的所有 HPS 复位输出接口。Altera 复位源 BFM 应用编程接口可应用于表 29 - 3 中列出的所有 BFM。

表 29 - 3. HPS 复位输出接口仿真模型

接口名称	BFM 实例名称
h2f_reset	h2f_reset
h2f_cold_reset	h2f_cold_reset
h2f_debug_apb_reset	h2f_debug_apb_reset

HPS 复位输出接口被连接到一个复位源 BFM。Qsys 配置表 29 - 4 中所示的 BFM。

表 29 - 4. HPS 复位输出接口的复位源 BFM 的配置

参数	BFM 值 (1)	含义
置位复位高电平	关闭	该参数为关闭状态，指定来自 BFM 的一个有效低复位信号。
初始复位的周期	0	该参数为 0，指定 BFM 不对复位信号自动置位。

表 29 - 4 注释:

(1) 对被例化的 BFM 的参数值的配置用于 HPS 仿真。

FPGA-to-HPS AXI 从接口

FPGA-to-HPS AXI 从接口，f2h_axi_slave 被连接到 Mentor Graphics AXI 的从器件 BFM 来进行仿真。Qsys 配置表 29-5 中所示的 BFM。BFM 时钟输入被连接到 f2h_axi_clock 时钟。

表 29-5. FPGA-to-HPS AXI 从器件 BFM 的配置

参数	值
AXI 地址宽度	32
AXI 读数据宽度	32、64、128
AXI 写数据宽度	32、64、128
AXI ID 宽度	8

您可以通过使用 BFM API 控制和监控 AXI 从器件 BFM。



要了解更多信息，请参考 *Mentor Verification IP Altera Edition User Guide*。要了解关于 FPGA-to-HPS AXI 从接口的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷 *HPS 组件接口* 章节中的“存储器映射的接口”。

HPS-to-FPGA AXI 主接口

HPS-to-FPGA AXI 主接口，h2f_axi_master 被连接到一个 Mentor Graphics AXI 主器件 BFM 进行仿真。Qsys 配置表 29-6 中所示的 BFM。BFM 时钟输入被连接到 h2f_axi_clock 时钟。

表 29-6. HPS-to-FPGA AXI 主器件 BFM 的配置

参数	值
AXI 地址宽度	30
AXI 读和写数据宽度	32、64、128
AXI ID 宽度	12

您可以通过使用 BFM API 控制和监控 AXI 主器件 BFM。



要了解更多信息，请参考 *Mentor Verification IP Altera Edition User Guide*。要了解关于 HPS-to-FPGA AXI 主接口的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *HPS 组件接口* 章节中的“存储器映射的接口”。


轻型 HPS-to-FPGA AXI 主接口

轻型 HPS-to-FPGA AXI 主接口，h2f_lw_axi_master 被连接到一个 Mentor Graphics AXI 主器件 BFM 来进行仿真。Qsys 配置表 29-7 中所示的 BFM。BFM 时钟输入被连接到 h2f_lw_axi_clock 时钟。

表 29-7. 轻型 HPS-to-FPGA AXI 主器件 BFM 的配置


参数	值
AXI 地址宽度	21
AXI 读和写数据宽度	32
AXI ID 宽度	12

可通过使用 BFM API 控制和监控 AXI 主器件 BFM。

 要了解更多信息，请参考 *Mentor Verification IP Altera Edition User Guide*。要了解关于轻型 HPS-to-FPGA AXI 主接口的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *HPS 组件接口* 章节中的“存储器映射的接口”。

FPGA-to-HPS SDRAM 接口

HPS 组件包含一个存储器接口仿真模型，所有的 FPGA-to-HPS SDRAM 接口都与其相连接。该模型基于 HPS 实现并且提供周期级的精确度，从而反映真实的带宽和接口的延迟。然而，模型不具有 HPS 软件提供的详细配置，因此，当使用不同的优先权或权重时，不会反映任何内端口调度（可能在实际硬件的冲突下出现的）。

 要了解更多信息，请参考 *外部存储器接口手册* 第 3 卷的 *Functional Description—Hard Memory Interface* 章节中的“SoC 器件中与 EMI 相关的 HPS 功能”。

HPS-to-FPGA MPU 通用 I/O 接口

HPS-to-FPGA MPU 通用 I/O 接口被连接到 Altera conduit BFM 以进行仿真。表 29-8 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29-8 中列出的 API 监测接口状态更改或设置接口。

表 29-8. HPS-to-FPGA MPU 通用 I/O 接口仿真模型

接口名称	BFM 实例名称	RTL 仿真 API 功能名称	Post-Fit 仿真 API 函数名称
h2f_mpu_gp	h2f_mpu_gp	set_h2f_mpu_gp_out() get_h2f_mpu_gp_in()	set_gp_out() get_gp_in()

HPS-to-FPGA MPU 事件接口

HPS-to-FPGA MPU 事件接口被连接到 Altera conduit BFM 以进行仿真。表 29-9 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29-9 中列出的 API 监控接口状态更改或设置接口。

表 29-9. HPS-to-FPGA MPU 事件接口仿真模型

接口名称	BFM 实例名称	RTL 仿真 API 函数名称 ⁽¹⁾	Post-Fit 仿真 API 函数名称
h2f_mpu_events	h2f_mpu_events	get_h2f_mpu_eventi() set_h2f_mpu_evento() set_h2f_mpu_standbywfe() set_h2f_mpu_standbywfi()	get_eventi() set_evento() set_standbywfe() set_standbywfi()

表 29-9 注释：
(1) conduit get_*() 和 set_*() API 函数的使用与一般 Avalon conduit BFM 的函数使用相同。

FPGA- to- HPS 中断接口

FPGA- to- HPS 中断接口被连接到 Altera Avalon 中断接收器 BFM 以进行仿真。
表 29 - 10 列出了每个接口的名称。

表 29 - 10. FPGA- to- HPS 中断接口仿真模型

接口名称	BFM 实例名称
f2h_irq0	f2h_irq0
f2h_irq1	f2h_irq1

Altera Avalon 中断接收器 BFM API 可应用于表 29 - 3 中列出的所有 BFM。

HPS- to- FPGA 调试 APB 接口

HPS- to- FPGA 调试 APB 接口被连接到 Altera conduit BFM 以进行仿真。表 29 - 11 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29 - 11 中列出的 API 函数监控接口状态更改或设置接口。

表 29 - 11. HPS- to- FPGA 调试 APB 接口仿真模型

接口名称	BFM 名称	RTL 仿真 API 函数名称	Post-Fit 仿真 API 函数名称
h2f_debug_apb	h2f_debug_apb	set_h2f_dbg_apb_PADDR()	set_PADDR()
		set_h2f_dbg_apb_PADDR_31()	set_PADDR_31()
		set_h2f_dbg_apb_PENABLE()	set_PENABLE()
		get_h2f_dbg_apb_PRDATA()	get_PRDATA()
		get_h2f_dbg_apb_PREADY()	get_PREADY()
		set_h2f_dbg_apb_PSEL()	set_PSEL()
		get_h2f_dbg_apb_PSLVERR()	get_PSLVERR()
		set_h2f_dbg_apb_PWDATA()	set_PWDATA()
		set_h2f_dbg_apb_PWRITE()	set_PWRITE()
h2f_debug_apb_sideband	h2f_debug_apb_sideband	get_h2f_dbg_apb_PCLKEN()	get_PCLKEN()
		get_h2f_dbg_apb_DBG_APB_DISABLE())	get_DBG_APB_DISABLE()

FPGA- to- HPS 系统跟踪宏单元 (STM) 硬件事件接口

FPGA- to- HPS STM 硬件事件接口被连接到 Altera conduit BFM 以进行仿真。表 29 - 12 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29 - 12 中列出的 API 函数监控接口状态更改或设置接口。

表 29 - 12. FPGA- to- HPS STM 硬件事件接口仿真模型

接口名称	BFM 名称	RTL 仿真 API 函数名称	Post-Fit 仿真 API 函数名称
f2h_stm_hw_events	f2h_stm_hw_events	get_f2h_stm_hwevents()	get_stm_events()

HPS- to-FPGA 交叉触发接口

HPS- to-FPGA 交叉触发接口被连接到 Altera conduit BFM 以进行仿真。表 29 - 13 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29 - 13 中列出的 API 函数监控接口状态更改或设置接口。

表 29 - 13. HPS- to-FPGA 交叉触发接口仿真模型

接口名称	BFM 名称	RTL 仿真 API 函数名称	Post-Fit 仿真 API 函数名称
h2f_cti	h2f_cti	get_h2f_cti_trig_in()	get_trig_in()
		set_h2f_cti_trig_in_ack()	set_trig_inack()
		set_h2f_cti_trig_out()	set_trig_out()
		get_h2f_cti_trig_out_ack()	get_trig_outack()
		set_h2f_cti_asicctl()	set_asicctl()
		get_h2f_cti_fpga_clk_en()	get_clk_en()

HPS- to-FPGA 跟踪端口接口

HPS- to-FPGA 跟踪端口接口被连接到 Altera conduit BFM 以进行仿真。表 29 - 14 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29 - 14 中列出的 API 函数监控接口状态更改或设置接口。

表 29 - 14. HPS- to-FPGA 跟踪端口接口仿真模型

接口名称	BFM 名称	RTL 仿真 API 函数名称	Post-Fit 仿真 API 函数名称
h2f_tpiu	h2f_tpiu	get_h2f_tpiu_clk_ctl()	get_traceclk_ctl()
		set_h2f_tpiu_data()	set_trace_data()

FPGA- to-HPS DMA 握手接口

FPGA- to-HPS DMA 握手接口被连接到 Altera conduit BFM 以进行仿真。表 29 - 15 列出了每个接口的名称，以及每个仿真类型的 API 函数名称。可以通过使用表 29 - 15 中列出的 API 监控接口状态更改或设置接口。

表 29 - 15. FPGA- to-HPS DMA 握手接口仿真模型 (1/2)

接口名称	BFM 实例名称	RTL 仿真 API 函数名称 ⁽¹⁾	Post-Fit 仿真 API 功能名称
f2h_dma_req0	f2h_dma_req0	get_f2h_dma_req0_req()	get_channel0_req()
		get_f2h_dma_req0_single()	get_channel0_single()
		set_f2h_dma_req0_ack()	set_channel0_xx_ack()
f2h_dma_req1	f2h_dma_req1	get_f2h_dma_req1_req()	get_channel1_req()
		get_f2h_dma_req1_single()	get_channel1_single()
		set_f2h_dma_req1_ack()	set_channel1_xx_ack()
f2h_dma_req2	f2h_dma_req2	get_f2h_dma_req2_req()	get_channel2_req()
		get_f2h_dma_req2_single()	get_channel2_single()
		set_f2h_dma_req2_ack()	set_channel2_xx_ack()

表 29 - 15 注释：
(1) conduit get_*() 和 set_*() API 函数的使用与一般 Avalon conduit BFM 函数的使用相同。


表 29 - 15. FPGA- to- HPS DMA 握手接口仿真模型 (2/2)

接口名称	BFM 实例名称	RTL 仿真 API 函数名称 ⁽¹⁾	Post-Fit 仿真 API 功能名称
f2h_dma_req3	f2h_dma_req3	get_f2h_dma_req3_req() get_f2h_dma_req3_single() set_f2h_dma_req3_ack()	get_channel3_req() get_channel3_single() set_channel3_xx_ack()
f2h_dma_req4	f2h_dma_req4	get_f2h_dma_req4_req() get_f2h_dma_req4_single() set_f2h_dma_req4_ack()	get_channel4_req() get_channel4_single() set_channel4_xx_ack()
f2h_dma_req5	f2h_dma_req5	get_f2h_dma_req5_req() get_f2h_dma_req5_single() set_f2h_dma_req5_ack()	get_channel5_req() get_channel5_single() set_channel5_xx_ack()
f2h_dma_req6	f2h_dma_req6	get_f2h_dma_req6_req() get_f2h_dma_req6_single() set_f2h_dma_req6_ack()	get_channel6_req() get_channel6_single() set_channel6_xx_ack()
f2h_dma_req7	f2h_dma_req7	get_f2h_dma_req7_req() get_f2h_dma_req7_single() set_f2h_dma_req7_ack()	get_channel7_req() get_channel7_single() set_channel7_xx_ack()

表 29 - 15 注释：
(1) conduit get_*() 和 set_*() API 函数的使用与一般 Avalon conduit BFM 函数的使用相同。

仿真流程

- 该部分介绍基于 HPS 设计的仿真流程。
- Altera 提供功能寄存器传输级 (RTL) 仿真和 post-fitter 门级仿真流程。仿真流程涉及以下主要步骤：
- 1. 例化 HPS 组件 — 请参考第 29 - 8 页的 “指定 Qsys 中的 HPS 仿真模型”。
 - 2. 生成 Qsys 中的系统，包括仿真模型 — 请参考第 29 - 11 页的 “在 Qsys 中生成 HPS 仿真模型”。
 - 3. 运行仿真 — 请参考以下部分的其中之一：
 - 第 29 - 11 页的 “运行 HPS RTL 仿真”
 - 第 29 - 12 页的 “运行 HPS Post-Fit 仿真”

 要了解关于仿真的一般信息，请参考 *Quartus II 手册* 第 3 卷的 *Simulating Altera Designs* 章节。

指定 Qsys 中的 HPS 仿真模型

- 以下步骤概括如何设置 HPS 组件以进行仿真。
- 1. 添加 Qsys Component Library 中的 HPS 组件。
 - 2. 通过选择或不选 HPS-FPGA 接口，根据您的应用需要配置组件。

- 将合适的 HPS 接口连接到系统中的其它组件。例如，将 FPGA-to-HPS AXI 从接口连接到系统中另一个组件的 AXI 主接口。

当创建组件时，确保 conduit 接口具有正确的角色名称、方向和宽度。表 29 - 16 列出了所有 HPS conduit 接口的角色名称、方向和宽度。



要了解关于将 HPS 组件添加到设计的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Instantiating the HPS Component* 章节。

表 29 - 16. HPS Conduit 接口 (1/3)

角色名称	方向	宽度
h2f_warm_reset_handshake		
h2f_pending_rst_req_n	输出	1
f2h_pending_rst_ack_n	输入	1
h2f_mpu_gp		
gp_in	输入	32
gp_out	输出	32
h2f_mpu_events		
eventi	输入	1
evento	输出	1
standbywfe	输出	2
standbywfi	输出	2
f2h_dma_req0		
req0_req	输入	1
req0_single	输入	1
req0_ack	输出	1
f2h_dma_req1		
req1_req	输入	1
req1_single	输入	1
req1_ack	输出	1
f2h_dma_req2		
req2_req	输入	1
req2_single	输入	1
req2_ack	输出	1
f2h_dma_req3		
req3_req	输入	1
req3_single	输入	1
req3_ack	输出	1

表 29 - 16. HPS Conduit 接口 (2/3)

角色名称	方向	宽度
f2h_dma_req4		
req4_req	输入	1
req4_single	输入	1
req4_ack	输出	1
f2h_dma_req5		
req5_req	输入	1
req5_single	输入	1
req5_ack	输出	1
f2h_dma_req6		
req6_req	输入	1
req6_single	输入	1
req6_ack	输出	1
f2h_dma_req7		
req7_req	输入	1
req7_single	输入	1
req7_ack	输出	1
h2f_debug_apb		
paddr	输入	18
paddr_31	输入	1
penable	输入	1
prdata	输出	32
pready	输出	1
psel	输入	1
pslverr	输出	1
pdata	输入	32
pwrite	输入	1
h2f_debug_apb_sideband		
pclken	输出	1
dbg_apb_disable	输出	1
f2h_stm_hw_events		
stm_hwevents	输出	28
h2f_cti		
trig_in	输出	8
trig_in_ack	输入	8
trig_out	输入	8
trig_out_ack	输出	8
asicctl	输入	8


表 29 - 16. HPS Conduit 接口 (3/3)

角色名称	方向	宽度
fpga_clk_en	输出	1
h2f_tpiu		
clk_ctl	输出	1
data	输入	32

在 Qsys 中生成 HPS 仿真模型


以下步骤概述如何生成仿真模型：

1. 进入 Qsys 中的 **Generation** 页。
2. 对于 RTL 仿真，请执行以下步骤：
 - a. 将 **Create simulation model** 设置为 Verilog.
 - b. 点击 **Generate**。

 HPS 仿真不支持 VHDL 仿真环境。

对于 post-fit 仿真，请执行以下步骤：

- a. 打开 **Create HDL design files for synthesis** 选项。
- b. 打开 **Create block symbol file (.bsf)** 选项。
- c. 点击 **Generate**。

 要了解关于生成 HPS 组件的一般信息，请参考 *Cyclone V 器件手册* 第 3 卷的 *Instantiating the HPS Component* 章节。要了解关于 Qsys 仿真的更多信息，请参考 *Quartus II 手册* 第 1 卷的 *Creating a System with Qsys* 章节中的“仿真一个 Qsys 系统”。

运行 HPS RTL 仿真

Qsys 对各种仿真器生成脚本，您可以使用它们完成仿真过程。表 29 - 17 为不同的供应商列出了仿真工具、脚本名称和脚本目录。

表 29 - 17. Qsys 对各种仿真器生成的脚本

仿真器	脚本名称	目录
Mentor Graphics Modelsim® Altera Edition	msim_setup.tcl	<project directory>/<Qsys design name>/simulation/mentor
Cadence NC-Sim	ncsim_setup.sh	<project directory>/<Qsys design name>/simulation/cadence
Synopsys VCS	vcs_setup.sh	<project directory>/<Qsys design name>/simulation/synopsys/vcs
Synopsys VCS-MX	vcsmx_setup.sh	<project directory>/<Qsys design name>/simulation/synopsys/vcsmx

 要了解仿真步骤的详细信息，请参考 *Mentor Verification IP Altera Edition User Guide*，和 *Qsys Tutorial* 章节的 *Avalon Verification IP Suite User Guide*。

运行 HPS Post-Fit 仿真

该部分介绍如何运行 HPS post-fit 仿真。Qsys 成功地生成后，执行以下步骤：

1. 通过执行以下步骤，将 Qsys 生成的综合文件集添加到 Quartus II 工程：
 - a. 在 Quartus II 软件中，点击 Assignments 菜单中的 **Settings**。
 - b. 在 **Settings - <your Qsys system name>** 对话框中，找到 **Files** 标签，浏览 *<your project directory>/<your Qsys system name>/synthesis/* 并且选择 *<your Qsys system name>.qip*。
 - c. 点击 **OK**。
 2. 如果需要，可以将包含 HPS 组件的 Qsys 系统例化为 Quartus II 工程顶层实体。
 3. 通过点击 Processing 菜单中的 **Start Compilation** 编译设计。
 4. 如果需要，通过执行以下步骤，更改 EDA Netlist Writer 设置：
 - a. 点击 Assignment 菜单中的 **Settings**。
 - b. 在 **Simulation** 标签的 **EDA Tool Settings** 标签下，可以指定以下 EDA Netlist Writer 设置：
 - **Tool name**— 仿真工具的名称
 - **Format for output netlist**
 - **Output directory**
 - c. 点击 **OK**。
- ❓ 要了解关于 EDA Netlist Writer 设置的更多信息，请参考 Quartus II Help 中的 *Simulation Page (Settings Dialog Box)*。
5. 要通过 Quartus II EDA Netlist Writer 创建 post-fitter 仿真模型，在 Start 菜单中，指向 **Processing** 并且点击 **Start EDA Netlist Writer**。
 6. 编译仿真工具中所需的仿真文件。表 29-18 列出了所需的库和文件。
 7. 开始仿真。

表 29-18. Post-Fit 仿真文件 (1/2)

库	目录 ⁽¹⁾	文件
Altera 验证 IP 库	<i><Avalon Verification IP>/lib/</i>	verbosity_pkg.sv avalon_mm_pkg.sv avalon_utilities_pkg.sv
Avalon 时钟源 BFM	<i><Avalon Verification IP>/altera_avalon_clock_source/</i>	altera_avalon_clock_source.sv

表 29-18 注释：

- (1) *<ACDS install>* = Altera Complete Design Suite 安装路径
<Avalon Verification IP> = *<ACDS install>/ip/altera/sopc_builder_ip/verification*
<AXI Verification IP> = *<ACDS install>/ip/altera/mentor_vip_ae*
<HPS Post-fit Sim> = *<ACDS install>/ip/altera/hps/postfitter_simulation*
<Device Sim Lib> = *<ACDS install>/quartus/eda/sim_lib*
- (2) 使用 Modelsim-Altera 时，不需要器件仿真库。
- (3) 在 Verilog HDL 和 VHDL 混合设计中需要混合语言仿真器。

表 29 - 18. Post-Fit 仿真文件 (2/2)

库	目录 ⁽¹⁾	文件
Avalon 复位源 BFM	<i><Avalon Verification IP>/altera_avalon_reset_source/</i>	altera_avalon_reset_source. sv
Avalon MM 从器件 BFM	<i><Avalon Verification IP>/altera_avalon_mm_slave_bfm/</i>	altera_avalon_mm_slave_bfm. sv
Avalon 中断接收器 BFM	<i><Avalon Verification IP>/altera_avalon_interrupt_sink/</i>	altera_avalon_interrupt_sink. sv
Mentor AXI 验证 IP 库	<i><AXI Verification IP>/common/</i>	questa_mvc_svapi.svh
Mentor AXI3 BFM	<i><AXI Verification IP>/axi3/bfm/</i>	mgc_common_axi.sv mgc_axi_master.sv mgc_axi_slave.sv
HPS Post-Fit 仿真库	<i><HPS Post-fit Sim>/</i>	All the files in the directory
器件仿真库 ⁽²⁾	<i><Device Sim Lib>/</i>	altera_primitives.v 220model.v sgate.v altera_mf.v altera_lnsim.sv cyclonev_atoms.v arriav_atoms.v mentor/cyclonev_atoms_ncrypt. t.v mentor/arriav_atoms_ncrypt. v
EDA Netlist Writer 生 成的 Post-Fit 仿真模型	<i><User project directory>/</i>	*.vo *.vho ⁽³⁾
用户测试台文件	<i><User project directory>/</i>	*.v *.sv *.vhd ⁽³⁾

表 29 - 18 注释:

- (1) *<ACDS install>* = Altera Complete Design Suite 安装路径
<Avalon Verification IP> = *<ACDS install>/ip/altera/sopc_builder_ip/verification*
<AXI Verification IP> = *<ACDS install>/ip/altera/mentor_vip_ae*
<HPS Post-fit Sim> = *<ACDS install>/ip/altera/hps/postfitter_simulation*
<Device Sim Lib> = *<ACDS install>/quartus/eda/sim_lib*
- (2) 使用 Modelsim-Altera 时, 不需要器件仿真库。
- (3) 在 Verilog HDL 和 VHDL 混合设计中需要混合语言仿真器。

对于 post-fit 仿真, 必须使用指定的层次调用测试编程中的 BFM API。层次格式是:

<DUT>.\<HPS>|fpga_interfaces|<interface><space>.<BFM>.<API function>

其中:

- *<DUT>* 是测试下的设计的实例名称, 在包含 HPS 组件的测试台中进行了例化。
- *<HPS>* 是 HPS 组件实例名称, 其在 Qsys 系统中使用。

- `<interface>` 是指定的 FPGA-to-HPS 或 HPS-to-FPGA 接口的实例名称。该名称可以在位于 `<project directory>/<Qsys design name>/synthesis/submodules` 的 `fpga_interfaces.sv` 文件中找到。
- `<space>`— 您必须在接口实例名称后插入一个空格。
- `<BFM>` 是 BFM 实例名称。在 `<ACDS install>/ip/altera/hps/postfitter_simulation` 中，识别与使用的接口类型相对应的 SystemVerilog 文件。SystemVerilog 文件包含 BFM 实例名称。

例如，轻型 HPS-to-FPGA 主接口层次的路径可以按如下所示形成：

```
top.dut.\my_hps_component|fpga_interface|hps2fpga_light_weight.h2f_lw_axi_master
```

请注意在 “hps2fpga_light_weight” 之后保持空格。省略该空格会导致仿真失败，因为包含空格的实例名称 “hps2fpga_light_weight ” 用于 Quartus® II 生成的 post-fit 仿真模型中。

文档修订历史

表 29 - 19 列出了该文档的修订历史。

表 29 - 19. 文档修订历史

日期	版本	修订内容
2012 年 11 月	1.1	<ul style="list-style-type: none">■ 添加了调试 APB、STM 硬件事件、FPGA 交叉触发、FPGA 跟踪端口接口。■ 添加了 post-fit 仿真的支持。■ 更新了一些 API 函数名称。■ 删除了 DMA 外设时钟。
2012 年 6 月	1.0	首次发布
2012 年 5 月	0.1	初稿